现代数学手册

·计算机数学卷

Modern Mathematics Handbook

《现代数学手册》编纂委员会

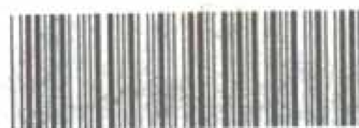
· 华中科技大学出版社 ·

现代数学手册

MODERN MATHEMATICS HANDBOOK

计算机数学卷

《现代数学手册》编纂委员会



· 华中科技大学出版社 ·
（华中理工大学出版社）
中国·武汉

《现代数学手册》编纂委员会

前　言

在人类开始跨入21世纪的历史时期,人们已普遍地看到了一种历史现象,即数学问题的多样性与数学应用的广泛性及深入性,已经成为现代科技发展的重要特征。可以预期,伴随着计算机科技在新世纪里的不断发展,此特征今后还将以更高的水平显示出来。

在中国,"科学技术是第一生产力"(邓小平名言)已逐渐成为人们信奉的朴实真理。国家富强显然要以第一生产力即科技的发达为必要条件。但是,如果没有近、现代发展起来的数学各分支学科作工具,当然也就不会有现代科技。因此"国家富强必须要依靠数学发达"这句经典名言(拿破仑(Napoleon)名言),自然也是一条不容置疑的客观真理。

基于上述认识,在华中理工大学出版社的倡议与委托下,我们通过集体协作,努力编纂了这部《现代数学手册》巨著,其目的正是怀着对我国将在新世纪里能尽快成为富强国家的热切希望,而欲为科技界提供一份力所能及的奉献。具体说来,这部工具性巨著服务的读者(或使用者)对象,包括广大科学工作者、工程技术人员、经济管理工作者、高等院校的教师和学生等。

那么,作为数学工具书,这部巨型手册要求具备哪些特点呢?在编写过程中,出版社负责人和我们达成了一项共识,即手册应具备科学性、先进性、实用性、规范性与简明性。200余位撰稿人与审稿人(来自中国科学院、北京大学、清华大学、复旦大学、南京大学、浙江大学、北京师范大学、厦门大学、上海交通大学、西安交通大学、中国科技大学、南开大学、武汉大学、华中理工大学、大连理工大学、南京航空航天大学、陕西师范大学等40多所高校与研究所)按照这些特点和要求付出了

艰辛的劳动。我们要感谢他们的通力合作与努力，使本手册基本上体现了上述所希冀的特点或特色。

为了读者选购和使用方便，本手册分5卷出版，分别名为“经典数学卷”、“近代数学卷”、“计算机数学卷”、“随机数学卷”和“经济数学卷”。需要指出的是，各个分支(篇目)的归属是相对的，这里考虑了各分卷篇幅大小的平衡问题。例如，“蒙特卡罗法”这一篇也可归入“计算机数学卷”。

我们要感谢诸分卷主编为精心组稿、编稿、审稿付出的精力和时间。特别要对中国科学院两位老院士钱伟长先生与吴文俊先生，以及杨叔子院士乐愿担任本手册的顾问而致以诚挚的谢忱。最后，还要对华中理工大学出版社具有远见卓识的负责人和埋头苦干的编辑人员与我们在本手册的生产全过程中的互相配合和精诚合作，深表谢忱。

《现代数学手册》编纂委员会

主编　徐利治

1999年12月于武汉

现代数学手册

篇目录

经典数学卷

近代数学卷

计算机数学卷

随机数学卷

经济数学卷

MODERN MATHEMATICS HANDBOOK

CONTENTS

CLASSICAL MATHEMATICS

MODERN MATHEMATICS

COMPUTER MATHEMATICS

STOCHASTIC MATHEMATICS

ECONOMIC MATHEMATICS

目　录

·计算机数学卷·

第1篇

数值分析

编　者　徐萃薇
审校者　高　立

目　录

引　言

数值分析是用数值计算的方法来研究数学分析中的一些问题.本篇内容包括：插值、拟合、数值微分、数值积分、常微分方程和偏微分方程的数值解法.其中除研究求解方程的数值方法外,也包括一些理论问题、如数值解的存在唯一性、格式的收敛性、稳定性以及误差分析等.

插值的使用可追溯到公元6世纪,当时中国科学家刘焯用等距二次插值法来计算天文学公式;到微积分创立的牛顿时代,对插值进行了进一步的研究.18世纪,欧拉用差商代替微商,开始了数值求解常微分方程初值问题的先例.1928年,柯朗等提出差分格式收敛的一个必要条件——CFL条件等等.但是,只是在20世纪后半叶电子计算机快速发展和普及以后,数值方法才得到了充分发展和广泛应用.

自然界中的各种现象,工程技术的不同领域,甚至人类社会活动的某些范畴(如政治、经济、文化和军事等),其规律性常常是用方程来表示的,求出方程的解,人们就可以定量以及定性地了解事物发展的规律和各种因素之间的制约关系.由于传统数学理论所提供方法的局限性,迫切要求有一种新的途径来求解这些问题,电子计算机的出现,使得今日用数值方法求解方程已成为主流.

1　误　差

1.1　误差的类型与来源

误差在近似计算中是不可缺少的,它主要产生于用数学和计算机来解决实际问题的过程中.误差有以下一些种类:

(1) 模型误差.用数学模型描述实在物理现象时要作简化,这种简化产生的误差叫**模型误差**.

(2) 观测误差.数学模型中通常会包含一些观测数据.这些观测数据不会绝对准确,这就会产生**观测误差**.如自由落体下落时,距离和时间的关系式

$$S(t)=\frac{1}{2}gt^2,$$

其中 g 是一个物理常数,通常取 $g\approx 9.81\mathrm{m/s^2}$,它是由观测得到的近似值.

(3) 截断误差.由模型求得的准确解与用数值方法求得的解之间的误差称**截断误差**.如一个无穷级数

$$\sum_{k=0}^{\infty}\frac{1}{k!}f^{(k)}(x_0),$$

在实际计算时,只能取前面有限项(如 n 项)

$$\sum_{k=0}^{n-1} \frac{1}{k!} f^{(k)}(x_0),$$

而
$$\sum_{k=n}^{\infty} \frac{1}{k!} f^{(k)}(x_0) = \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(x_0) - \sum_{k=0}^{n-1} \frac{1}{k!} f^{(k)}(x_0)$$
就是截断误差.

(4) 舍入误差.因计算机字长有限,原始数据在计算机上表示会产生误差,这个误差称舍入误差.如 π、$\sqrt{2}$、1/3 等,在计算机上只能取有限位(如取小数后四位),则

$$\rho_1 = 3.1416 - \pi = +0.0000074\cdots,$$
$$\rho_2 = 1.4142 - \sqrt{2} = -0.000013\cdots,$$
$$\rho_3 = 0.3333 - \frac{1}{3} = -0.000033\cdots$$

就是舍入误差.

1.2　误差的一些基本概念

(1) 浮点数.任何一个浮点数均可以表示为

$$\pm \beta^J w, \tag{1-1}$$

其中 β 叫做基,如十进制数,基 $\beta = 10$,二进制数,基 $\beta = 2$;J 称为阶,是一个整数,取正、负或零;w 称为尾数,由 t 位小数构成,可表示为

$$w = 0.\alpha_1\alpha_2\cdots\alpha_t,$$

其中 $1 \leqslant \alpha_1 \leqslant \beta - 1, 0 \leqslant \alpha_i \leqslant \beta - 1 (i = 2,3,\cdots,t)$,尾数中 t 叫做浮点数的精度.

(2) 误差.设 x^* 是准确值 x 的一个近似值,则 x^* 和 x 之差称为误差,用 e 表示:

$$e = x^* - x.$$

误差可正可负,误差为正,x^* 称为强近似;误差为负,x^* 称为弱近似.

(3) 误差界.若事先估计出误差的绝对值不超过某个正数 ε,则 ε 称为 x^* 的误差界,即

$$|e| = |x^* - x| \leqslant \varepsilon.$$

用 $x = x^* \pm \varepsilon$ 表示 x^* 的精确度,即准确值 x 所在的范围,亦即

$$x^* - \varepsilon \leqslant x \leqslant x^* + \varepsilon.$$

(4) 有效数字.将 x^* 表示成

$$x^* = \pm 0.\alpha_1\alpha_2\cdots\alpha_n \times 10^p, \tag{1-2}$$

其中 $\alpha_1 \neq 0$, p 是一整数.若其误差界

$$|x^* - x| \leqslant \frac{1}{2} \times 10^{p-n},$$

则 x^* 具有 n 位有效数字.

(5) 相对误差.称

$$e_r = \frac{e}{x} = \frac{x^* - x}{x}$$

为 x^* 的**相对误差**.由于 x 一般是未知的,而且 x 与 x^* 相差不大,所以也用

$$e_r = \frac{e}{x^*} = \frac{x^* - x}{x^*}$$

表示 x^* 的相对误差.

(6) 相对误差界.相对误差绝对值的上界叫作相对误差界,用 ε_r 表示,即

$$\varepsilon_r = \frac{\varepsilon}{|x^*|}.$$

形如(1-2)式的近似数 x^*,具有 n 位有效数字,则其相对误差界

$$\varepsilon_r = \frac{\varepsilon}{|x^*|} \leqslant \frac{1}{2\alpha_1}10^{-(n-1)}. \tag{1-3}$$

但要注意的是,形如(1-2)式的近似数 x^*,当相对误差界满足关系式

$$\varepsilon_r \leqslant \frac{1}{2(\alpha_1 + 1)}10^{-(n-1)} \tag{1-4}$$

时,x^* 至少具有 n 位有效数字.

1.3 误差分析

1.3.1 基本算术运算结果的误差界

设 x^* 和 y^* 分别表示 x 和 y 的近似值,并把它们的误差界看作是相应的微分,即

$$\mathrm{d}x = |x^* - x|; \quad \mathrm{d}y = |y^* - y|,$$

则

$$\begin{cases} \mathrm{d}(x \pm y) = \mathrm{d}x + \mathrm{d}y, \\ \mathrm{d}(xy) \approx |y|\,\mathrm{d}x + |x|\,\mathrm{d}y, \\ \mathrm{d}\dfrac{x}{y} \approx \dfrac{|x|\,\mathrm{d}y + |y|\,\mathrm{d}x}{|y|^2} \quad (y \neq 0). \end{cases} \tag{1-5}$$

若把 $\mathrm{d}_r x$ 与 $\mathrm{d}_r y$ 看作是 x^* 和 y^* 的相对误差界,即

$$\mathrm{d}_r x = \frac{\mathrm{d}x}{x} = \mathrm{d}\ln x; \quad \mathrm{d}_r y = \frac{\mathrm{d}y}{y} = \mathrm{d}\ln y,$$

则

$$\begin{cases} \mathrm{d}_r(x + y) \approx \max(\mathrm{d}_r x, \mathrm{d}_r y) \quad (x, y \text{ 同号}), \\ \mathrm{d}_r(x - y) \approx (|x|\,\mathrm{d}_r x + |y|\,\mathrm{d}_r y)/|x - y| \quad (x, y \text{ 同号}), \\ \mathrm{d}_r(x \cdot y) \approx \mathrm{d}_r x + \mathrm{d}_r y, \\ \mathrm{d}_r \dfrac{x}{y} \approx \mathrm{d}_r x + \mathrm{d}_r y \quad (y \neq 0). \end{cases} \tag{1-6}$$

1.3.2 函数求值的误差估计

在计算函数值$f(x)$时，由于自变量x不精确，会使$f(x)$产生误差。若x的近似值为x^*，用$f(x^*)$表示$f(x)$的近似值，误差界$df(x)$可用泰勒(Taylor)公式估计。假设f在包含x和x^*的一个开区间上存在足够高阶导数，则有

$$df(x) = f(x^*) - f(x) = f'(x^*)(x^* - x) + \frac{f''(\xi)}{2!}(x^* - x)^2,$$

其中$\xi \in (x, x^*)$。取绝对值得

$$|df(x)| = |f(x^*) - f(x)| \leqslant |f'(x^*)| dx + \left|\frac{f''(\xi)}{2!}\right|(dx)^2.$$

若$f''(x)$与$f'(x)$相比不太大，则可忽略高阶项得

$$df(x) \approx |f'(x^*)| dx. \tag{1-7}$$

如果$|f'(x^*)|$是零或值很小，则要考虑后面的项，特别，若

$$f'(x^*) = f''(x^*) = \cdots = f^{(k-1)}(x^*) = 0,$$
$$f^{(k)}(x^*) \neq 0,$$

且$|f^{(k+1)}(\xi)|$不很大，$\xi \in (x, x^*)$，则

$$df(x^*) \approx \left|\frac{f^{(k)}(x^*)}{k!}\right|(dx)^k. \tag{1-8}$$

对多元函数的误差界可用多元函数的泰勒公式得到

$$df(x_1, \cdots, x_n) \approx \sum_{i=1}^{n}\left|\frac{\partial f(x_1, \cdots, x_n)}{\partial x_i}\right| dx_i. \tag{1-9}$$

2 插　　值

表 2-1

x	x_0	x_1	$\cdots$	x_n
y	y_0	y_1	$\cdots$	y_n

若通过某种方法已知$y = f(x)$在区间$[a, b]$上的一组对应关系如表2-1所示，插值的目的是根据给定数据，寻找一个解析函数$\varphi(x)$近似地代替$f(x)$。

函数$\varphi(x)$的类型可以有不同选择，但最常用的是代数多项式，用多项式做插值函数称为**代数插值**。

2.1 代数插值的提法及存在唯一性

在$[a, b]$区间上给出了$n+1$个点上的函数值表2-1以后，需要构造一个多项式$\varphi(x)$满足条件：

1° $\varphi(x)$是不超过n次的多项式；

2° $\varphi(x_i) = f(x_i) = y_i \quad (i = 0, 1, \cdots, n)$.

$\varphi(x)$称为**插值函数**，x_i称为**插值节点**，条件1°、2°称为**插值条件**，$[a, b]$称为**插值区**

间.

定理 1　满足插值条件的插值多项式 $\varphi(x)$ 存在而且唯一.

2.2　拉格朗日插值

拉格朗日(Lagrange)插值是构造插值函数的一种方法,需首先引进基函数.

2.2.1　基函数

克罗内克尔(Kronecker)符号为

$$\delta_{ij}=\begin{cases}1, & i=j,\\ 0, & i\neq j.\end{cases} \tag{2-1}$$

基函数 $l_i(x)$ 的取值应满足条件:

$$l_i(x_k)=\delta_{ik},\quad i,k=0,1,\cdots,n. \tag{2-2}$$

它的表达形式为

$$\begin{aligned} l_i(x)&=\frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}\\ &=\prod_{\substack{j=0\\ j\neq i}}^{n}\frac{x-x_j}{x_i-x_j},\qquad i=0,1,\cdots,n. \end{aligned} \tag{2-3}$$

$l_i(x),i=0,1,\cdots,n$,称为 $x_0,x_1,\cdots,x_n$ 点上的 n 次插值基函数.

特别地,对 $n=1$,插值基函数为

$$l_0(x)=\frac{x-x_1}{x_0-x_1},$$

$$l_1(x)=\frac{x-x_0}{x_1-x_0},$$

其图形如图 2-1.

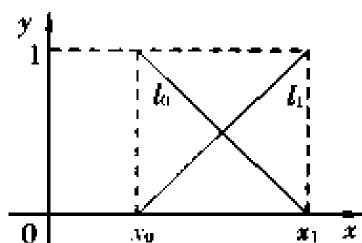


图 2-1

对 $n=2$,插值基函数为

$$l_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)},$$

$$l_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},$$

$$l_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)},$$

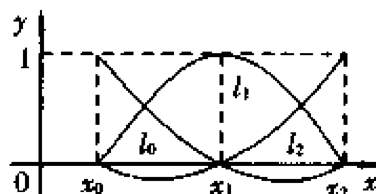


图 2-2

其图形如图 2-2.

2.2.2　拉格朗日插值多项式及余项

离散数据表 2-1 的 n 次拉格朗日插值多项式 $\varphi_n(x)$ 是插值基函数 $l_i(x)(i=0,$

$1,\cdots,n$) 的线性组合,即

$$\varphi_n(x)=\sum_{i=0}^{n}y_i l_i(x)=\sum_{i=0}^{n}\frac{\omega(x)}{(x-x_i)\omega'(x_i)}y_i, \tag{2-4}$$

其中

$$\begin{cases}\omega(x)=(x-x_0)(x-x_1)\cdots(x-x_n),\\ \omega'(x_i)=(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n).\end{cases} \tag{2-5}$$

定理2 设 $\varphi_n(x)$ 是过点 $x_0,x_1,\cdots,x_n$ 的 n 次插值多项式,$f(x)\in C^n[a,b]$,$f^{(n+1)}(x)$ 在$[a,b]$上存在,其中$[a,b]$是包含 $x_0,x_1,\cdots,x_n$ 的任一区间,则对任意给定的 $x\in[a,b]$,总存在一点 $\xi\in(a,b)$(依赖于 x) 使

$$R_n(x)=f(x)-\varphi_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(x), \tag{2-6}$$

其中 $\omega(x)$ 由(2-5)式定义.

特别地,对 $n=1$ 的情形,$\varphi_1(x)$ 称为**线性插值**,

$$\varphi_1(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1,$$

余项

$$R_1(x)=f(x)-\varphi_1(x)=\frac{f''(\xi)}{2!}(x-x_0)(x-x_1),\quad \xi\in(a,b),$$

$[a,b]$是包含 x_0、x_1 的任一区间.由于 $|(x-x_0)(x-x_1)|$ 在 $x=(x_0+x_1)/2$ 达到最大值,因此有

$$|R_1(x)|\leqslant\frac{(x_1-x_0)^2}{8}\max_{a\leqslant x\leqslant b}|f''(x)|. \tag{2-7}$$

2.3 分段线性插值

2.3.1 龙格现象

并不是插值多项式次数越高,对函数的逼近就越好,一个有名的实例是

例1 给定函数

$$f(x)=\frac{1}{1+25x^2},\quad -1\leqslant x\leqslant 1,$$

取等距节点 $x_i=-1+\frac{2}{10}i$ ($i=0,1,\cdots,10$),构造 $\varphi_{10}(x)$:

$$\varphi_{10}(x)=\sum_{i=0}^{10}f(x_i)l_i(x),$$

其中

$$l_i(x)=\frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_{10})}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_{10})}.$$

计算结果见图2-3.从图2-3可以看出,插值多项式 $\varphi_{10}(x)$ 在$[-1,1]$上并不收敛于

$f(x)$,这就是有名的龙格(Runge)现象.

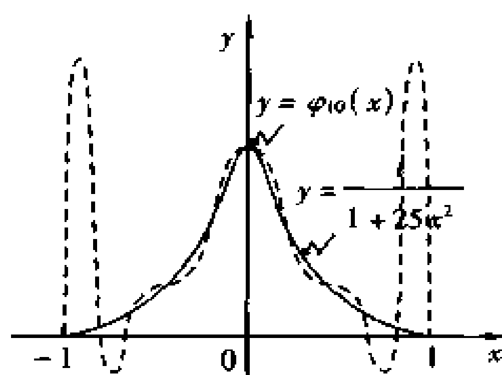


图 2-3

2.3.2 构造分段线性插值函数

给定离散数据表 2-1,作连续函数 $\varphi(x)$ 满足下面条件:

1° $\varphi(x)$ 在每个小区间 $[x_i, x_{i+1}]$ 上是线性函数;

2° $\varphi(x_i) = y_i, i = 0,1,\cdots,n$.

分段线性插值基函数 $l_i(x)$ 的图形如图 2-4.

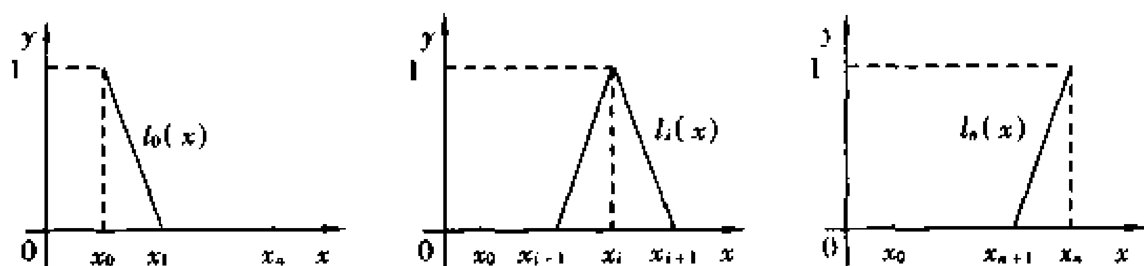


图 2-4

表达式为

$$
\begin{cases}
l_0(x) = \begin{cases} \dfrac{x - x_1}{x_0 - x_1}, & x_0 \leqslant x \leqslant x_1, \\ 0, & x_1 < x \leqslant x_n, \end{cases} \\
l_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leqslant x \leqslant x_i, \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}}, & x_i < x \leqslant x_{i+1}, \\ 0, & [a,b] - [x_{i-1}, x_{i+1}], i = 1,2,\cdots,n-1, \end{cases} \\
l_n(x) = \begin{cases} \dfrac{x - x_{n-1}}{x_n - x_{n-1}}, & x_{n-1} \leqslant x \leqslant x_n, \\ 0, & x_0 \leqslant x < x_{n-1}. \end{cases}
\end{cases}
$$

显然,分段线性插值基函数满足条件(2-2),分段线性插值函数的表达式为

$$\varphi(x) = \sum_{i=0}^{n} y_i l_i(x). \tag{2-8}$$

2.3.3 分段线性插值函数的误差估计

定理 3 设 $\varphi(x)$ 是过点 $x_0, x_1, \cdots, x_n$ 的分段线性插值函数，$f(x) \in C[a,b]$，$f''(x)$ 在 $[a,b]$ 上存在，则

$$|R(x)| = |f(x) - \varphi(x)| \leqslant \frac{h^2}{8} M, \tag{2-9}$$

其中 $h = \max\limits_{0 \leqslant i \leqslant n-1} |x_{i+1} - x_i|$，$M = \max\limits_{a \leqslant x \leqslant b} |f''(x)|$.

2.4 埃尔米特插值

埃尔米特(Hermite)插值也叫带导数的插值，即不但在给定节点上取已知函数值，而且取已知的导数值.

2.4.1 n 个节点上的 $2n+1$ 次埃尔米特插值

(1) 给定 $n+1$ 个节点上相应的函数值和导数值如表 2-2 所示.

表 2-2

x	x_0	x_1	x_2	$\cdots$	x_n
y	y_0	y_1	y_2	$\cdots$	y_n
y'	m_0	m_1	m_2	$\cdots$	m_n

要求构造一个 $2n+1$ 次的埃尔米特插值多项式 $H(x)$ 满足下面条件：

1° $H(x)$ 是不超过 $2n+1$ 次的多项式；

2° $H(x_i) = y_i, H'(x_i) = m_i; i = 0,1,\cdots,n.$ (2-10)

为此先构造 $2n+2$ 个插值基函数 $\alpha_i(x), \beta_i(x), i = 0,1,\cdots,n$，$\alpha_i(x)$ 和 $\beta_i(x)$ 的取值满足条件

$$\begin{cases} \alpha_i(x_j) = \delta_{ij}, & \alpha'_i(x_j) = 0, \\ \beta_i(x_j) = 0, & \beta'_i(x_j) = \delta_{ij}, \end{cases} \quad i,j = 0,1,\cdots,n, \tag{2-11}$$

其中 δ_{ij} 由(2-1)式定义. 由条件(2-11)可直接推出

$$\begin{cases} \alpha_i(x) = \left(1 - 2(x - x_i)\sum\limits_{\substack{j=0 \\ j \neq i}}^{n} \dfrac{1}{x_i - x_j}\right) l_i^2(x), \\ \beta_i(x) = (x - x_i) l_i^2(x), \quad i = 0,1,\cdots,n, \end{cases} \tag{2-12}$$

其中 $l_i(x)$ 由(2-3)式定义. 则 $2n+1$ 次埃尔米特插值多项式为

$$H_{2n+1}(x) = \sum_{i=0}^{n} (y_i \alpha_i(x) + m_i \beta_i(x)). \tag{2-13}$$

(2) 埃尔米特插值的余项估计有

定理 4　设 $H_{2n+1}(x)$ 是通过 $x_0,x_1,\cdots,x_n$ 的 $2n+1$ 次埃尔米特插值多项式，$f(x)\in C^{2n+1}[a,b]$，$f^{(2n+2)}(x)$ 在 $[a,b]$ 上存在，$[a,b]$ 是包含 $x_0,x_1,\cdots,x_n$ 的任一区间，则对于任意给定的 $x\in[a,b]$，总存在 $\xi\in(a,b)$（依赖于 x），使

$$R_{2n+1}(x)=f(x)-H_{2n+1}(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}\omega^2(x), \tag{2-14}$$

其中 $\omega(x)$ 由(2-5)式定义.

(3) 当 $n=1$ 时，三次埃尔米特插值多项式的节点为 x_0、x_1，插值基函数为

$$\alpha_0(x)=\left(1+2\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_1}{x_0-x_1}\right)^2,$$

$$\alpha_1(x)=\left(1+2\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_0}{x_1-x_0}\right)^2,$$

$$\beta_0(x)=(x-x_0)\left(\frac{x-x_1}{x_0-x_1}\right)^2,$$

$$\beta_1(x)=(x-x_1)\left(\frac{x-x_0}{x_1-x_0}\right)^2.$$

其图形如图 2-5 所示.

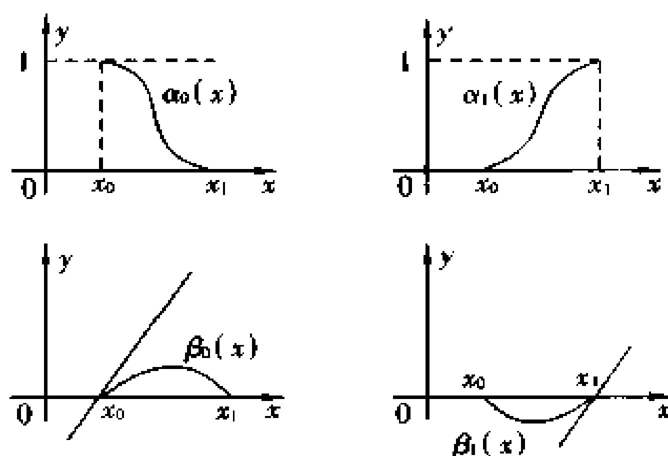


图 2-5

三次埃尔米特插值多项式的表达式为

$$H_3(x)=y_0\alpha_0(x)+y_1\alpha_1(x)+m_0\beta_0(x)+m_1\beta_1(x), \tag{2-15}$$

余项 $R_3(x)$ 为

$$R_3(x)=\frac{f^{(4)}(\xi)}{4!}(x-x_0)^2(x-x_1)^2,\quad \xi\in(a,b), \tag{2-16}$$

其中 $[a,b]$ 为包含 x_0,x_1 的任一区间.

2.4.2　分段三次埃尔米特插值

给定离散数据表 2-2，要求构造一个插值多项式 $H(x)$ 满足下面条件：

1° 在每个小区间 $[x_i,x_{i+1}]$ 上是三次多项式；

2° $H(x_i)=y_i$，$H'(x_i)=m_i$，$i=0,1,\cdots,n$；

3° $H(x) \in C[a,b]$.

$2n+2$ 个插值基函数 $\alpha_i(x)$、$\beta_i(x)(i=0,1,\cdots,n)$ 的定义是

$$\alpha_0(x) = \begin{cases} \left(1 + 2\dfrac{x - x_0}{x_1 - x_0}\right)\left(\dfrac{x - x_1}{x_0 - x_1}\right)^2, & x_0 \leqslant x \leqslant x_1, \\ 0, & x_1 < x \leqslant x_n, \end{cases}$$

$$\alpha_i(x) = \begin{cases} \left(1 + 2\dfrac{x - x_i}{x_{i-1} - x_i}\right)\left(\dfrac{x - x_{i-1}}{x_i - x_{i-1}}\right)^2, & x_{i-1} \leqslant x \leqslant x_i, \\ \left(1 + 2\dfrac{x - x_i}{x_{i+1} - x_i}\right)\left(\dfrac{x - x_{i+1}}{x_i - x_{i+1}}\right)^2, & x_i < x \leqslant x_{i+1}, \\ 0, & [a,b] - [x_{i-1}, x_{i+1}], \end{cases} \quad i = 1,2,\cdots,n-1$$

$$\alpha_n(x) = \begin{cases} \left(1 + 2\dfrac{x - x_n}{x_{n-1} - x_n}\right)\left(\dfrac{x - x_{n-1}}{x_n - x_{n-1}}\right)^2, & x_{n-1} \leqslant x \leqslant x_n, \\ 0, & x_0 \leqslant x < x_{n-1}, \end{cases}$$

$$\beta_0(x) = \begin{cases} (x - x_0)\left(\dfrac{x - x_1}{x_0 - x_1}\right)^2, & x_0 \leqslant x \leqslant x_1, \\ 0, & x_1 < x \leqslant x_n, \end{cases}$$

$$\beta_i(x) = \begin{cases} (x - x_i)\left(\dfrac{x - x_{i-1}}{x_i - x_{i-1}}\right)^2, & x_{i-1} \leqslant x \leqslant x_i, \\ (x - x_i)\left(\dfrac{x - x_{i+1}}{x_i - x_{i+1}}\right)^2, & x_i < x \leqslant x_{i+1}, \\ 0, & [a,b] - [x_{i-1}, x_{i+1}], \end{cases} \quad i = 1,2,\cdots,n-1,$$

$$\beta_n(x) = \begin{cases} (x - x_n)\left(\dfrac{x - x_{n-1}}{x_n - x_{n-1}}\right)^2, & x_{n-1} \leqslant x \leqslant x_n, \\ 0, & x_0 \leqslant x < x_{n-1}. \end{cases}$$

分段三次埃尔米特插值多项式的表达式为

$$H(x) = \sum_{i=0}^{n}(y_i\alpha_i(x) + m_i\beta_i(x)). \tag{2-17}$$

其余项估计有

定理 5 设 $H(x)$ 是 $a = x_0 < x_1 < \cdots < x_n = b$ 上的分段三次埃尔米特插值多项式，$f(x) \in C^3[a,b]$，$f^{(4)}(x)$ 在$[a,b]$上存在，则对任一给定的 $x \in [a,b]$，总存在 $\xi \in (a,b)$，使

$$|R(x)| = |f(x) - H(x)| \leqslant \frac{h^4}{384}M_4, \tag{2-18}$$

其中 $h = \max\limits_{0 \leqslant i \leqslant n-1}|x_{i+1} - x_i|$，$M_4 = \max\limits_{a \leqslant x \leqslant b}|f^{(4)}(x)|$.

2.5 三次样条插值

2.5.1 问题的提出

三次样条是应用最广泛的样条，它有明确的力学意义. 开始是绘图员绘图时用有弹性的木棒固定在若干样点上，然后画出光滑的曲线. 在数学上可描述为

定义 1 设在区间$[a,b]$上给定一个划分：

$$a = x_0 < x_1 < \cdots < x_n = b,$$

$S(x)$是以$x_0, x_1, \cdots, x_n$为样点的**三次样条函数**，如果$S(x)$满足条件：

1° $S(x_i) = y_i, \quad i = 0,1,\cdots,n$;

2° 在每个小区间$[x_i, x_{i+1}]$上，$S(x)$是三次多项式；

3° $S(x) \in C^2[a,b]$.

由定义 1 知，$S(x)$可供利用的条件共$4n-2$个，即$n+1$个插值条件和$S(x) \in C^2[a,b]$给出的$3n-3$个连续性条件：$S(x_i-0) = S(x_i+0)$，$S'(x_i-0) = S'(x_i+0)$，$S''(x_i-0) = S''(x_i+0)$，$i = 1,2,\cdots,n-1$. 但$S(x)$是由n段次数不超过 3 的多项式组成，共有$4n$个待定系数，因此，要唯一确定$S(x)$还缺少两个条件，通常在区间的两个端点附加两个边界条件. 附加的边界条件一般有三种：

第 Ⅰ 类边界条件　两端点斜率已知：

$$S'(x_0) = y_0' = m_0, \quad S'(x_n) = y'_n = m_n.$$

第 Ⅱ 类边界条件　两端点处的二阶导数已知：

$$S''(x_0) = y_0'' = M_0, \quad S''(x_n) = y''_n = M_n.$$

特殊情况，$S''(x_0) = S''(x_n) = 0$称为**自然边界条件**.

第 Ⅲ 类边界条件　$f(x)$是周期函数，即在端点上满足

$$S'(x_0) = S'(x_n), \quad S''(x_0) = S''(x_n).$$

定理6 对第 Ⅰ、第 Ⅱ、第 Ⅲ 类边界条件，三次样条函数的解均存在而且唯一.

2.5.2 三次样条插值函数的构造

令$S'(x_i) = m_i (i = 0,1,\cdots,n)$，再在每个小区间$[x_i, x_{i+1}]$上构造三次埃尔米特插值多项式. 利用(2-15)式得到.

$$S_i(x) = \left(1 + 2\frac{x - x_i}{x_{i+1} - x_i}\right)\left(\frac{x - x_{i+1}}{x_i - x_{i+1}}\right)^2 y_i + \left(1 + 2\frac{x - x_{i+1}}{x_i - x_{i+1}}\right)\left(\frac{x - x_i}{x_{i+1} - x_i}\right)^2 y_{i+1} +$$

$$(x - x_i)\left(\frac{x - x_{i+1}}{x_i - x_{i+1}}\right)^2 m_i + (x - x_{i+1})\left(\frac{x - x_i}{x_{i+1} - x_i}\right)^2 m_{i+1}. \tag{2-19}$$

为计算m_i，对$S_i(x)$求导并利用样点上二阶导数的连续性，即$S''_i(x_i-0) = S''_i(x_i+0)$得

$$\frac{6}{h_{i-1}^2} y_{i-1} - \frac{6}{h_{i-1}^2} y_i + \frac{2}{h_{i-1}} m_{i-1} + \frac{4}{h_{i-1}} m_i$$

$$= -\frac{6}{h_i^2}y_i + \frac{6}{h_i^2}y_{i+1} - \frac{4}{h_i}m_i - \frac{2}{h_i}m_{i+1},$$

其中 $h_i = x_{i+1} - x_i$. 整理后得

$$(1 - a_i)m_{i-1} + 2m_i + a_i m_{i+1} = b_i, \quad i = 1,2,\cdots,n-1, \tag{2-20}$$

其中

$$\begin{cases} a_i = \dfrac{h_{i-1}}{(h_{i-1} + h_i)}, \\ b_i = 3\left(\dfrac{1 - a_i}{h_{i-1}}(y_i - y_{i-1}) + \dfrac{a_i}{h_i}(y_{i+1} - y_i)\right). \end{cases}$$

方程(2-20) 是关于 $n+1$ 个未知数 m_i 的 $n-1$ 个方程，要求解还须补充两个方程.

对第 Ⅰ 类边界条件，因 m_0 和 m_n 已知，所以方程组(2-20)中只有 $n-1$ 个未知数，可直接求解.

对第 Ⅱ 类边界条件，由 $S''_1(x_0+0)$ 和 $S''_n(x_n-0)$ 已知，可给出两个方程：

$$\begin{cases} 2m_0 + m_1 = \dfrac{3}{h_0}(y_1 - y_0) - \dfrac{h_0}{2}M_0, \\ m_{n-1} + 2m_n = \dfrac{3}{h_{n-1}}(y_n - y_{n-1}) + \dfrac{h_{n-1}}{2}M_n. \end{cases} \tag{2-21}$$

(2-20) 和(2-21) 式联立，即可解出 m_i.

对第 Ⅲ 类边界条件，有

$$\begin{cases} m_0 = m_n, \\ \dfrac{3}{h_0^2}(y_1 - y_0) - \dfrac{1}{h_0}(2m_0 + m_1) = \dfrac{3}{h_{n-1}^2}(y_{n-1} - y_n) + \dfrac{1}{h_{n-1}}(m_{n-1} + 2m_n), \end{cases} \tag{2-22}$$

(2-20) 和(2-22) 式联立，也可解出 m_i.

三次样条插值的计算步骤如下：

(1) 根据给定的 (x_i, y_i) 及相应的边界条件，计算方程组(2-20) 及相应边界条件的系数 a_i, b_i.

(2) 解方程组求 m_i.

(3) 用求得的 m_i 代入(2-19) 式，给出小区间上的三次插值函数 $S_i(x)$.

(4) 计算区间 $[a,b]$ 上的样条函数 $S(x)$.

2.5.3 三次样条插值的收敛性

定理 7 设 $f(x) \in C^4[a,b]$，给定划分

$$\Delta: \quad a = x_0 < x_1 < \cdots < x_n = b,$$

并设 $S_\Delta(x)$ 是对划分 Δ 的样条插值函数，则当

$$\delta = \max_{0 \le i \le n-1} |x_{i+1} - x_i| \to 0$$

时，对一切 $x \in [a,b]$ 恒有

$$|f^{(i)}(x) - S_\Delta(x)| \leqslant c_i \delta^{4-i}, \quad i = 0,1,2,3, \tag{2-23}$$

其中 c_i 是与划分 Δ 无关的常数.

样条函数不一定必须逐段是三次多项式,也可以逐段是一个简单函数,连续点保持足够光滑.但三次多项式计算简单,且满足一般实际问题的要求,故用得较多.

3　曲线拟合

3.1　曲线拟合及最小二乘原理

3.1.1　基本原理

在实验或统计中,人们需要从一组测定数据(如 m 个点(x_i, y_i))去求自变量 x 和应变量 y 的一个近似解析表达式 $y = \varphi(x)$,这就是在 m 个点$(x_i, y_i)(i = 1,2,\cdots,m)$上求数据拟合的问题.它与前面介绍的插值不同.插值要求得到的函数 $\varphi(x)$ 通过所有插值点(x_i, y_i),即要求曲线满足条件 $y_i = \varphi(x_i)$.但在实验中测得的数据一般都有误差,要求曲线 $\varphi(x)$ 必须通过点(x_i, y_i)不但会把误差保留下来,有时也不一定能反映数据的真实规律.

数据拟合则是根据测定数据间的相互关系,确定拟合曲线 $y = \varphi(x; a_0, a_1, \cdots, a_n)$ 的类型,然后根据在给定点上误差的平方和达到最小的原则,即用

$$F(a_0, a_1, \cdots, a_n) = \sum_{i=1}^{m}[\varphi(x_i; a_0, a_1, \cdots, a_n) - y_i]^2 = \min$$

定出参数 $a_k^*(k = 0,1,\cdots,n)(n < m)$,从而得到**拟合曲线** $y^* = \varphi(x)$.

下面用一个例子来说明曲线拟合的过程.

例1　给出一组实验数据如表3-1.

表 3-1

	1	2	3	4	5
x	2	4	5	8	9
y	1.3	4	5.5	7	8

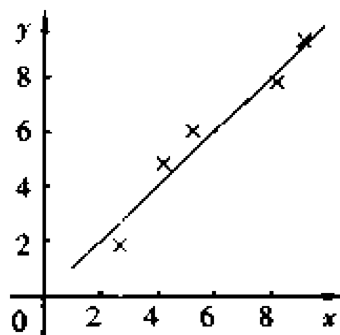


图 3-1

将点标在坐标纸上,如图 3-1.容易看出这些点在一条直线附近.因此选用直线方程

$$\varphi(x) = a_0 + a_1 x$$

为拟合曲线.令

$$F(a_0, a_1) = \sum_{i=1}^{5}((a_0 + a_1 x_i) - y_i)^2,$$

要使 F 达到最小，从数学分析知道，应求解方程组

$$\begin{cases}\dfrac{\partial F}{\partial a_0}=2\sum\limits_{i=1}^{5}(a_0+a_1x_i-y_i)=0,\\ \dfrac{\partial F}{\partial a_1}=2\sum\limits_{i=1}^{5}x_i(a_0+a_1x_i-y_i)=0.\end{cases}$$

这个方程组称为**正规方程**.化简并将数据代入得

$$\begin{cases}5a_0+28a_1=25.8,\\ 28a_0+190a_1=174.1,\end{cases}$$

解之得，$a_0^*=0.165$，$a_1^*=0.892$，于是

$$y^*=\varphi(x)=0.165+0.892x.$$

总括以上过程，求拟合曲线的步骤为：

步1　由观测数据画粗略图形——散点图.

步2　由散点图确定拟合曲线类型.

步3　用最小二乘原理，确定函数中的参数.

3.1.2　一般情形的最小二乘拟合

若给出一组实验数据 $(x_i,y_i)(i=1,2,\cdots,m)$，并选定拟合曲线方程为 $y=\varphi(x;a_0,a_1,\cdots,a_n)$，$n<m$. 令

$$F(a_0,a_1,\cdots,a_n)=\sum_{i=1}^{m}\omega_i(\varphi(x_i;a_0,a_1,\cdots,a_n)-y_i)^2,$$

其中 $\omega_i>0(i=1,2,\cdots,m)$ 是权系数.最小二乘拟合是求参数 $a_k^*(k=0,1,\cdots,n)$，使 $F(a_0,a_1,\cdots,a_n)$ 达到最小.由多元函数求极值的必要条件知，应

$$\frac{\partial F}{\partial a_k}=0\quad(k=0,1,\cdots,n).$$

当 F 为 a_i 的非线性函数时，称为**非线性最小二乘拟合**，如果 F 为 a_k 的线性函数，则称为**线性最小二乘拟合**.对线性最小二乘拟合，可令

$$\varphi(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\cdots+a_n\varphi_n(x),\tag{3-1}$$

其中 $\varphi_i(x)(i=0,1,\cdots,n)$ 是线性无关的，而且

$$F(a_0,a_1,\cdots,a_n)=\sum_{i=1}^{m}\omega_i(\sum_{k=0}^{n}a_k\varphi_k(x_i)-y_i)^2.\tag{3-2}$$

对 F 求偏导数得

$$\frac{\partial F}{\partial a_j}=2\sum_{i=1}^{m}\omega_i(\sum_{k=0}^{n}a_k\varphi_k(x_i)-y_i)\varphi_j(x_i)=0,\tag{3-3}$$

$$j=0,1,\cdots,n.$$

若记

$$y_i=f(x_i)=\varphi(x_i;a_0,a_1,\cdots,a_n),$$

$$(\varphi_k,\varphi_j)=\sum_{i=1}^{m}\omega_i\varphi_k(x_i)\varphi_j(x_i),$$

$$(f,\varphi_j) = \sum_{i=1}^{m}\omega_i f(x_i)\varphi_j(x_i)$$

(3-3) 式可写成

$$\sum_{k=0}^{n}(\varphi_k,\varphi_j)a_k = (f,\varphi_j),\quad j = 0,1,\cdots,n. \tag{3-4}$$

方程(3-4) 是正规方程,是关于 $a_0,\cdots,a_n$ 的 $n+1$ 阶线性方程组,它的系数矩阵是

$$\boldsymbol{\Phi} = \begin{bmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n) \\ \vdots & \vdots & & \vdots \\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n) \end{bmatrix}, \tag{3-5}$$

由于 $\{\varphi_0,\varphi_1,\cdots,\varphi_n\}$ 线性无关,故 $\det\boldsymbol{\Phi}\neq 0$,方程组(3-4) 存在唯一解 $a_0^*,a_1^*,\cdots,a_n^*$. 因而可以得到离散数据 (x_i,y_i) 的最小二乘拟合曲线

$$y^* = a_0^*\varphi_0(x) + a_1^*\varphi_1(x) + \cdots + a_n^*\varphi_n(x). \tag{3-6}$$

若取 $\varphi_k(x) = x^k (k = 0,1,\cdots,n)$,这时得到的最小二乘拟合曲线为

$$y^* = a_0^* + a_1^* x + \cdots + a_n^* x^n.$$

3.1.3 非线性曲线的数据拟合

在许多实际问题中,变量相互之间的内在关系并不像前面说的那样简单呈线性关系,这时我们可以把拟合曲线中的自变量 x 和应变量 y 看成是其它变量的函数. 如原来的函数关系是

$$f(\tilde{y}) = a + bg(\tilde{x}),$$

令 $x = g(\tilde{x}), y = f(\tilde{y})$,经过变换后得到

$$y = a + bx,$$

这样把原来非线性问题转化成了线性问题.

另外,如何寻找符合实际情况的拟合曲线,也是非常重要的,这一方面要根据专业知识和经验来选择,另一方面也要根据散点图的形状来选择.

例 2 已知一组实验数据如表 3-2.

表 3-2

t	1	2	3	4	5	6	7	8
$y\times 10^{-3}$	4.00	6.40	8.00	8.80	9.22	9.50	9.70	9.80
t	9	10	11	12	13	14	15	16
$y\times 10^{-3}$	10.00	10.20	10.32	10.42	10.50	10.55	10.59	10.60

将数据标在坐标纸上画出散点图如图 3-2. 从图上看到,开始时 y 增加较快,后逐渐变缓,到一定时间后基本稳定在一个数值上. 根据这些特点,选双曲线

$$\frac{1}{y} = a + \frac{b}{t},\text{即 } y = \frac{t}{at+b} \tag{3-7}$$

作为拟合曲线. 令 $\tilde{y} = 1/y, x = 1/t$, 则(3-7) 式可改写成

$$\tilde{y} = a + bx.$$

利用表 3-3 的数据,用前面同样的办法计算出正规方程为

$$\begin{cases} 16a + 3.38073b = 1.8372 \times 10^3, \\ 3.38073a + 1.58435b = 0.52886 \times 10^3, \end{cases}$$

解之得 $a = 80.6621, b = 161.6822$. 从而得到拟合曲线

$$y = \frac{t}{80.6621t + 161.6822} = S^{(1)}(t)$$

各点上的误差 $\delta_i^{(1)} = S^{(1)}(t_i) - y_i (i = 1,2,\cdots,16)$.

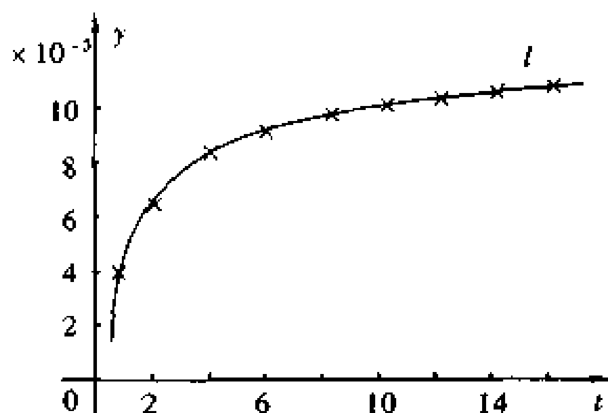


图 3-2

由图 3-2 还可以选用指数曲线

$$y = ae^{b/t}$$

作为拟合曲线. 先两边取对数得

$$\ln y = \ln a + \frac{b}{t},$$

令 $\tilde{y} = \ln y, A = \ln a, x = \frac{1}{t}$, 则得

$$\tilde{y} = A + bx.$$

和前面一样计算出 $A = -4.48072, b = -1.0567$, 于是拟合曲线为

$$y = 11.3253 \times 10^{-3} e^{-1.0567/t} = S^{(2)}(t).$$

各点上的误差

$$\delta_i^{(2)} = S^{(2)}(t_i) - y_i \quad (i = 1,2,\cdots,16).$$

对同一个问题,我们选用了两种不同类型的曲线拟合,哪种拟合更好呢?原则是选误差小的. 对于这个例题,由计算知

$$\max_i |\delta_i^{(1)}| = 0.568 \times 10^{-3},$$

$$\max_i |\delta_i^{(2)}| = 0.277 \times 10^{-3},$$

可见, $\|\delta^{(2)}\|_\infty$ 比 $\|\delta^{(1)}\|_\infty$ 小,因此,选 $S^{(2)}(t)$.

用最小二乘法做曲线拟合时,数学模型选择要通过实际计算才能确定,目前许多计算机配有自动选择数学模型的软件.

3.2 多变量的数据拟合

若影响应变量的因素不只一个,就是多变量数据拟合问题,前面介绍的最小二乘数据拟合的有关概念和结论,都可以直接推广到多变量的情形.

若已知一组正数 $\{\omega_i\}(i = 1,2,\cdots,m)$ 和多元函数 $y = f(x_1,\cdots,x_l)$, 并给出一组测量得到的数据表 3-3.

表 3-3

编号	x_1	x_2	…	x_l	y
1	x_{11}	x_{12}	…	x_{1l}	y_1
2	x_{21}	x_{22}	…	x_{2l}	y_2
⋮	⋮	⋮		⋮	⋮
x_m	x_{m1}	x_{m2}	…	x_{ml}	y_m

一般说来 $m > l$. 如果选择的拟合曲线方程为

$$P(x_1,\cdots,x_l) = \sum_{k=0}^{n} a_k\varphi_k(x_1,\cdots,x_l), \tag{3-8}$$

和前面一样,用最小二乘原理来确定方程(3-8)中的全部系数 a_k. 为此,令

$$F(a_0,\cdots,a_n) = \sum_{i=1}^{m}\omega_i(P(x_{1i},\cdots,x_{li}) - y_i)^2,$$

为使 F 达到极小,求解

$$\frac{\partial F}{\partial a_k} = 0, \quad k = 0,1,\cdots,n.$$

和前面单变量情形一样,权系数 $\{\omega_i\}$ 和系数 $a_0,a_1,\cdots,a_n$ 仍满足正规方程(3-4),只是这里的

$$(\varphi_k,\varphi_j) = \sum_{i=1}^{m}\omega_i\varphi_j(x_{1i},\cdots,x_{li})\varphi_k(x_{1i},\cdots,x_{li}).$$

解正规方程(3-4)得到系数 a_k^*,于是

$$P(x_1,\cdots,x_l) = \sum_{k=0}^{n} a_k^*\varphi_k(x_1,\cdots,x_l),$$

称为函数 $y = f(x_1,\cdots,x_l)$ 的最小二乘拟合曲线.

3.3 用正交多项式作最小二乘拟合

用一般多项式作最小二乘拟合时,其正规方程往往是病态的,为了避免出现病态方程,通常采用正交化方法. 对给定点集 $\{x_i\}(i = 1,\cdots,m)$ 及权系数 $\{\omega_i\}(i = 1,\cdots,m)$,如果函数序列 $\{\varphi_i(x)\}(i = 0,\cdots,n)$ 满足

$$(\varphi_k,\varphi_j) = \sum_{i=1}^{m}\omega_i\varphi_k(x_i)\varphi_j(x_i) = \begin{cases}0, & j \neq k,\\ A_k > 0, & j = k,\end{cases} \tag{3-9}$$

则称 $\{\varphi_j\}$ 是关于点集 $\{x_i\}$ 带权 $\{\omega_i\}$ 正交.

当 $\{\varphi_j\}$ 正交时,正规方程(3-4)的求解就非常容易,得

$$a_k^* = \frac{(f,\varphi_k)}{(\varphi_k,\varphi_k)} = \frac{\sum_{i=1}^{m}\omega_i f(x_i)\varphi_k(x_i)}{\sum_{i=1}^{m}\omega_i\varphi_k^2(x_i)}, \tag{3-10}$$

$$k = 0,1,\cdots,n.$$

问题是如何根据给定数据 $x_1,x_2,\cdots,x_m$ 及权系数 $\omega_1,\cdots,\omega_m$ 去构造带权的正交多项式序列 $\{P_i\}$. 下面给出递推关系：

$$\begin{cases} P_0(x) = 1, \\ P_1(x) = (x - a_1)P_0(x), \\ P_{k+1}(x) = (x - a_{k+1})P_k(x) - \beta_k P_{k-1}(x), \end{cases} \tag{3-11}$$

$$k = 1,2,\cdots,n-1,\ n < m,$$

这里 $P_k(x)$ 是首项系数为1的 k 次多项式,待定系数 a_k 及 β_k 可根据 $\{P_k(x)\}$ 的正交性求得：

$$\begin{cases} a_{k+1} = \dfrac{(xP_k, P_k)}{(P_k, P_k)}, \\ \beta_k = \dfrac{(P_k, P_k)}{(P_{k-1}, P_{k-1})}, \end{cases} \quad k = 1,2,\cdots,n-1. \tag{3-12}$$

用归纳法可以证明这样给出的 $\{P_k(x)\}$ 是正交的. 用正交多项式 $\{P_k(x)\}$ 的线性组合作最小二乘拟合,只要在根据(3-11)式和(3-12)式逐步求 $P_k(x)$ 的同时用(3-10)式计算系数 a_k^*. 于是所求的拟合曲线

$$y^* = a_0^* P_0(x) + a_1^* P_1(x) + \cdots + a_n^* P_n(x).$$

用这个方法求拟合曲线不用解方程组,只用递推公式,是一个较好的算法,一般的数学库中有用这个算法编制的标准程序供用户调用.

4 数值积分与数值微分

数值积分是用数值计算的方法求定积分

$$I(f) = \int_a^b f(x)\mathrm{d}x$$

的值. 这似乎很简单,只要求出 $f(x)$ 的原函数 $F(x)$,积分值 $I(f) = F(b) - F(a)$ 就知道了. 但实际是

(1) 有很多简单积分的原函数无法用初等函数表示.

(2) $f(x)$ 本身是用表格形式给出的.

因此,讨论数值积分是非常必要的.

4.1 牛顿-科茨公式、梯形求积公式、抛物线求积公式

4.1.1 牛顿-科茨(Newton-Cotes)公式

(1) 把区间$[a,b]$ n 等分,设其分点为

$$x_i = a + ih,\quad i = 0,1,\cdots,n, h = \frac{b-a}{n}.$$

过这 $n+1$ 个节点，构造一个 n 次插值多项式

$$P_n(x) = \sum_{i=0}^{n} \frac{\omega(x)}{(x-x_i)\omega'(x_i)} f(x_i),$$

其中 $\omega(x)$ 和 $\omega'(x_i)$ 由(2-5)式给出．用 $P_n(x)$ 代替 $f(x)$，则有

$$I(f) = \int_a^b f(x)\mathrm{d}x \approx \int_a^b P_n(x)\mathrm{d}x$$

$$= \sum_{i=0}^{n}\left(\int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)}\mathrm{d}x\right) f(x_i) = \sum_{i=0}^{n} A_i f(x_i), \tag{4-1}$$

其中
$$A_i = \int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)}\mathrm{d}x.$$

对 A_i 中的积分变量 x 作变量替换，令 $x = a + th$，可得

$$A_i = (b-a)c_i^{(n)}, \tag{4-2}$$

其中
$$c_i^{(n)} = \frac{(-1)^{n-i}}{n(i!)((n-i)!)}\int_0^n \frac{t(t-1)\cdots(t-n)}{(t-i)}\mathrm{d}t. \tag{4-3}$$

(4-1)式叫作**牛顿－科茨公式**，(4-3)式叫**牛顿－科茨系数**．下面给出 $i = 1,2,\cdots,6$ 的牛顿－科茨系数表(表4-1)．

表 4-1

n	$c_i^{(n)}$						
1	$\frac{1}{2}$	$\frac{1}{2}$					
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$				
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$			
4	$\frac{7}{90}$	$\frac{16}{45}$	$\frac{2}{15}$	$\frac{16}{45}$	$\frac{7}{90}$		
5	$\frac{19}{288}$	$\frac{25}{96}$	$\frac{25}{144}$	$\frac{25}{144}$	$\frac{25}{96}$	$\frac{19}{288}$	
6	$\frac{41}{840}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{34}{105}$	$\frac{9}{280}$	$\frac{9}{35}$	$\frac{41}{840}$

(2) 牛顿－科茨公式的误差估计有

定理 1　1° 如果 n 为偶数，$f^{(n+2)}(x)$ 在 $[a,b]$ 上连续，则有

$$E_n(f) = c_n h^{n+3} f^{(n+2)}(\xi),\quad \xi \in [a,b], \tag{4-4}$$

其中
$$c_n = \frac{1}{(n+2)!}\int_0^n t^2(t-1)\cdots(t-n)\mathrm{d}t.$$

2° 如果 n 为奇数，$f^{(n+1)}(x)$ 在 $[a,b]$ 连续，则

$$E_n(f) = c_n h^{n+2} f^{(n+1)}(\xi),\quad \xi \in [a,b], \tag{4-5}$$

其中
$$c_n = \frac{1}{(n+1)!}\int_0^n t(t-1)\cdots(t-n)\mathrm{d}t.$$

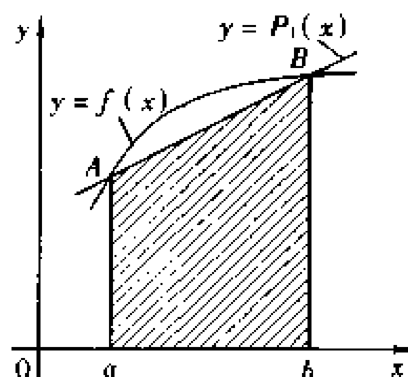


图 4-1

从定理1知，如果 $f(x)$ 是 n 次多项式，则有 $f^{(n+1)}(x)=0$，即 $E_n(f)=0$. 也就是说，对次数不高于 n 次的多项式，牛顿-科茨公式(4-1)精确成立.

(3) 代数精确度：

定义 1　如果求积公式(4-1)对所有次数不高于 n 次的多项式精确成立，但对 $n+1$ 次多项式等式不精确成立，则称求积公式具有 n 次**代数精确度**.

4.1.2　梯形求积公式

特别取牛顿-科茨公式中的 $n=1$，则有 $c_0^{(1)}=c_1^{(1)}=1/2$,

$$I(f)\approx I_1(f)=\frac{b-a}{2}(f(a)+f(b)). \tag{4-6}$$

公式(4-6)叫做**梯形求积公式**. 这是因为从几何上看是用通过 $A(a,f(a))$、$B(b,f(b))$ 两点的直线 $P_1(x)$ 逼近 $f(x)$，用梯形面积 $AabB$ 代替曲边梯形面积，见图4-1. 其误差估计有

定理 2　若 $f(x)$ 在 $[a,b]$ 上有二阶连续导数，则梯形求积公式(4-6)有

$$E_1(f)=I(f)-I_1(f)$$
$$=-\frac{(b-a)^3}{12}f''(\xi),\quad \xi\in[a,b]. \tag{4-7}$$

梯形求积公式的代数精确度为1.

4.1.3　抛物线求积公式

取牛顿-科茨公式中的 $n=2$，则有 $c_0^{(2)}=c_2^{(2)}=1/6$, $c_1^{(2)}=4/6$,

$$I(f)\approx I_2(f)=\frac{h}{3}(f(a)+4f(\frac{a+b}{2})+f(b)), \tag{4-8}$$

其中 $h=(b-a)/2$. 公式(4-8)叫作**抛物线求积公式**，也叫**辛普森(Simpson)公式**. 这是因为通过三点作抛物线，用抛物线围成的曲边梯形面积来代替 $f(x)$ 围成的曲边梯形面积，见图4-2. 公式(4-8)的误差估计有

定理3　若 $f(x)$ 在 $[a,b]$ 上有四阶连续导数，则有

$$E_2(f)=I(f)-I_2(f)=-\frac{1}{90}h^5f^{(4)}(\xi),$$
$$\xi\in[a,b]. \tag{4-9}$$

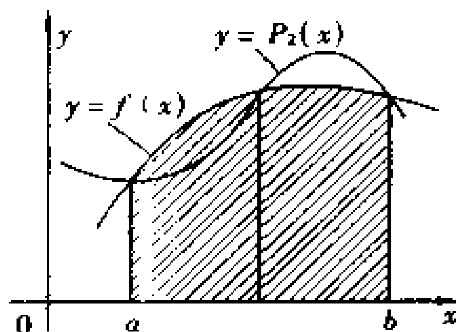


图 4-2

抛物线求积公式的代数精确度为3.

三阶以上的牛顿-科茨公式称为高阶牛顿-科茨公式，虽然其精度高，但由于舍入误差干扰较大，在实际计算中用得不多.

4.2　复化求积公式

在大区间上用梯形求积公式或抛物线求积公式，精度较低，用高阶牛顿 - 科茨公式有舍入误差干扰，均不理想. 因此，我们采用把积分区间分小，在每个小区间上用精度不很高的求积公式，这就是复化求积公式的基本思想.

4.2.1　复化梯形公式

把区间$[a,b]$ n 等分：

$$x_i = a + ih,\quad i = 0,1,\cdots,n, h = \frac{b-a}{n},$$

对每个小区间$[x_i, x_{i+1}]$用梯形求积公式，则

$$I(f) = \int_a^b f(x)\mathrm{d}x = \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)\mathrm{d}x$$

$$\approx \frac{h}{2}(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(a+ih)) = T_n, \tag{4-10}$$

T_n 称为**复化梯形求积公式**，下标 n 表示将区间 n 等分. 复化梯形求积公式的误差估计有

定理 4　若 $f(x) \in C^2[a,b]$，则

$$R(f, T_n) = -\frac{b-a}{12}h^2 f''(\xi),\quad \xi \in [a,b]. \tag{4-11}$$

4.2.2　复化抛物线求积公式

因为抛物线求积公式用到了区间的中点，所以在构造复化抛物线求积公式时必须把区间$[a,b]$等分为偶数份. 为此，令 $n = 2m$，m 为正整数，在每个小区间$[x_{2i-2}, x_{2i}]$上用抛物线求积公式，则

$$I(f) = \int_a^b f(x)\mathrm{d}x = \sum_{i=1}^{m}\int_{x_{2i-2}}^{x_{2i}} f(x)\mathrm{d}x$$

$$\approx \frac{h}{3}[f(a) + f(b) + 4\sum_{i=1}^{m} f(x_{2i-1}) + 2\sum_{i=1}^{m-1} f(x_{2i})] = S_n, \tag{4-12}$$

其中 $h = (b-a)/n$. S_n 称为**复化抛物线求积公式**. 复化抛物线求积公式的误差估计有

定理 5　设 $f(x) \in C^4[a,b]$，则

$$R(f, S_n) = -\frac{b-a}{2\,880}(2h)^4 f^{(4)}(\xi),\quad \xi \in [a,b]. \tag{4-13}$$

定义 2　设复化求积公式为 $I(f) \approx I_n$，如果当 $h \to 0$ 时有

$$\frac{I(f) - I_n}{h^p} \longrightarrow c,$$

其中 c 是一非零常数，则称 I_n 是 **p 阶收敛的**.

复化梯形求积公式是二阶收敛,复化抛物线求积公式是四阶收敛.

4.3　逐次分半法

对给定的积分问题,选定了求积公式后,确定区间应分成多少子区间能满足计算的要求?亦即 n 应取多大?可以根据误差估计来求 n.这一方面要求高阶导数,有一定困难,而且求得的 n 一般都比较保守.

逐次分半法是根据精度要求,在计算过程中把区间逐次分半,并利用前面计算的结果来判断是否达到精度要求.

4.3.1　复化梯形求积公式的逐次分半算法

由复化梯形求积公式的误差估计(4-11)可以推出,

$$I(f) - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n), \tag{4-14}$$

这就提供了误差判断的条件.如果

$$|T_{2n} - T_n| < \varepsilon \quad (\varepsilon \text{ 是允许误差}),$$

则 T_{2n} 就是满足精度要求的结果.要注意的是如何充分利用老分点上的函数值.现将具体计算步骤列示于下:

步1　取 $n = 1$,计算

$$T_1 = (b-a)\left(\frac{f(a)}{2} + \frac{f(b)}{2}\right).$$

步2　取 $n = 2$,计算

$$T_2 = \frac{b-a}{2}\left(\frac{f(a)}{2} + \frac{f(b)}{2} + f(x_1)\right) = \frac{T_1}{2} + \frac{b-a}{2}f(x_1).$$

步3　取 $n = 4$,计算

$$T_4 = \frac{b-a}{4}\left(\frac{f(a)}{2} + \frac{f(b)}{2} + f(x_1) + f(x_2) + f(x_3)\right) = \frac{T_2}{2} + \frac{b-a}{4}(f(x_1) + f(x_3)).$$

步4　对一般情形有计算公式

$$T_{2n} = \frac{T_n}{2} + \frac{b-a}{2n}\sum_{i=1}^{n} f\left(a + (2i-1)\frac{b-a}{2n}\right). \tag{4-15}$$

步5　如果 $|T_{2n} - T_n| < \varepsilon$,计算停止.

4.3.2　抛物线求积公式的逐次分半法

和前面同样处理,将区间逐次分半,由(4-13)式可以推出:

$$I(f) - S_{2n} \approx \frac{1}{15}(S_{2n} - S_n). \tag{4-16}$$

计算序列 $S_1, S_2, \cdots, S_n, S_{2n}, \cdots$,其中

$$S_n = \frac{h_n}{2}(f(a) + f(b) + 2S_n^{(1)} + 4S_n^{(2)}), \quad n = 1,2,\cdots,$$

$$h_n = \frac{b-a}{2n},$$

$S_n^{(1)}$ 是在原来分点上的函数值之和,即

$$S_n^{(1)} = f(x_2) + f(x_4) + \cdots + f(x_{2n-2}),$$

$S_n^{(2)}$ 是在新分点上函数值之和,即

$$S_n^{(2)} = f(x_1) + f(x_3) + \cdots + f(x_{2n-1}),$$

检验条件为

$$| S_{2n} - S_n | < \epsilon \quad (\epsilon \text{ 为允许误差}).$$

满足条件,计算停止,则 S_{2n} 为要求的近似解.

4.4　理查森外推法和龙贝格求积法

4.4.1　理查森(Richardson)外推法

假设用一个步长为 h 的函数 $F_1(h)$ 去逼近一个数 F^*. 对 $F_1(h)$ 的截断误差有估计式

$$R(F^*) = F^* - F_1(h) = a_1 h^{p_1} + a_2 h^{p_2} + \cdots + a_k h^{p_k} + \cdots, \tag{4-17}$$

其中 $p_k > p_{k-1} > \cdots > p_1 > 0, a_i (i = 1,2,\cdots)$ 都是与 h 无关的常数. 也就是说 $F_1(h)$ 逼近 F^* 的阶是 h^{p_1}. 现在构造一个新的序列

$$F_2(h) = \frac{1}{1-q^{p_1}}(F_1(qh) - q^{p_1}F_1(h)), \quad 1 - q^{p_1} \neq 0,$$

则

$$F^* - F_2(h) = F^* - \frac{1}{1-q^{p_1}}(F_1(qh) - q^{p_1}F_1(h))$$
$$= a_2^{(2)} h^{p_2} + a_3^{(2)} h^{p_3} + \cdots + a_k^{(2)} h^{p_k} + \cdots,$$

其中 $a_2^{(2)}, a_3^{(2)}, \cdots, a_k^{(2)}, \cdots$ 都是与 h 无关的常数. $F_2(h)$ 逼近于 F^* 的阶为 h^{p_2}. 重复上面做法,可得到

$$F_{m+1}(h) = \frac{1}{1-q^{p_m}}(F_m(qh) - q^{p_m}F_m(h)), \tag{4-18}$$

其中 q 满足 $1 - q^{p_m} \neq 0$. 称(4-18)式为**理查森外推法**.

定理 6　如果 F 逼近 F^* 的误差由(4-17)式给出,那么(4-18)式逼近于 F^* 的误差为 $h^{p_{m+1}}$.

理查森算法的步骤见表 4-2.

表 4-2

① $F_1(h)$				
② $F_1(qh)$	③ $F_2(h)$			
④ $F_1(q^2h)$	⑤ $F_2(qh)$	⑥ $F_3(h)$		
⑦ $F_1(q^3h)$	⑧ $F_2(q^2h)$	⑨ $F_3(qh)$	⑩ $F_4(h)$	
⋮	⋮	⋮	⋮	…

4.4.2　龙贝格积分法

龙贝格(Romberg)求积是在复化梯形求积公式的基础上应用理查森外推构造出来的一种算法.复化梯形求积公式的误差可以表示为

$$R(f,T_n)=a_2h^2+a_4h^4+a_6h^6+\cdots. \tag{4-19}$$

其中 $a_2,a_4,a_6,\cdots$ 都是与 h 无关的常数;在复化梯形求积公式中把 T_n 记为 $T(h)$,h 为步长,将步长缩小一半后得到的复化梯形求积公式记为 $T(h/2)$,这样就可以得到一个序列

$$T(h),T\left(\frac{h}{2}\right),T\left(\frac{h}{2^2}\right),\cdots,$$

显然序列收敛到积分值 $I(f)$.利用(4-19)式及理查森外推公式(4-18),取 $q=1/2$ 可得算法

$$\begin{cases}T_1(h)=T(h),\\ T_{m+1}(h)=\dfrac{T_m(h/2)-(1/2)^{2m}T_m(h)}{1-(1/2)^{2m}}, & m=1,2,\cdots,\end{cases} \tag{4-20}$$

(4-20)式称为**龙贝格积分公式**. $T_{m+1}(h)$ 逼近 $I(f)$ 的误差为

$$I(f)-T_{m+1}(h)=a_{m+1}^{(m+1)}h^{2(m+1)}+a_{m+2}^{(m+1)}h^{2(m+2)}+\cdots,$$

其中 $a_{m+1}^{(m+1)},a_{m+2}^{(m+1)},\cdots$ 是与 h 无关的常数.令

$$T_m^{(k)}=T_m\left(\frac{b-a}{2^k}\right),$$

则计算过程可列成表4-3,表中的每一列与对角线都收敛到定积分 $I(f)$.

表 4-3

$T_1^{(0)}$				
$T_1^{(1)}$	$T_2^{(0)}$			
$T_1^{(2)}$	$T_2^{(1)}$	$T_3^{(0)}$		
$T_1^{(3)}$	$T_2^{(2)}$	$T_3^{(1)}$	$T_4^{(0)}$	
⋮	⋮	⋮	⋮	…

现将龙贝格的计算步骤叙述于下:

步1　计算 $T_1^{(0)}=\dfrac{b-a}{2}(f(a)+f(b))$.

对 $l=1,2,\cdots$ 计算下列各步：

步 2　计算

$$T_1^{(l)}=\frac{T_1^{(l-1)}}{2}+\frac{b-a}{2^l}\sum_{j=1}^{2^{l-1}}f\left(a+(2j-1)\frac{b-a}{2^l}\right).$$

步 3　计算表 4-2 中的第 $l+1$ 行元素

$$T_{m+1}^{(k-1)}=\frac{4^m T_m^{(k)}-T_m^{(k-1)}}{4^m-1},\qquad \begin{aligned}m&=1,2,\cdots,l\\k&=l,l-1,\cdots.\end{aligned}$$

步 4　收敛控制：如果表 4-2 对角线上最后两个相邻元素之差满足精度要求，即

$$|T_m^{(0)}-T_{m-1}^{(0)}|<\varepsilon,$$

则 $T_m^{(0)}$ 作为近似值，计算停止，否则 l 增加 1 后，重复步骤 2,3,4.

4.5　高斯型求积公式

定义 3　对于积分

$$I=\int_a^b\rho(x)f(x)\mathrm{d}x,$$

其中 $\rho(x)$ 是 $[a,b]$ 上的权函数，适当选择节点 $x_1,x_2,\cdots,x_n$，使求积公式

$$\int_a^b\rho(x)f(x)\mathrm{d}x\approx\sum_{k=1}^{n}A_kf(x_k)\tag{4-21}$$

具有 $2n-1$ 次代数精确度，则称(4-21) 式为**高斯型求积公式**.

高斯型求积公式的主要问题是如何在 $[a,b]$ 上选 n 个不等距节点 $x_1,x_2,\cdots,x_n$，使 $f(x)$ 是不超过 $2n-1$ 次多项式时等式(4-21) 精确成立.

令　$$\omega_n(x)=(x-x_0)(x-x_1)\cdots(x-x_n),$$

用 $\omega_n(x)$ 去除 $f(x)$，并表示为

$$f(x)=q(x)\omega_n(x)+r(x),$$

其中 $q(x)$ 和 $r(x)$ 都是不超过 $n-1$ 次的多项式. 于是有

$$\int_a^b\rho(x)f(x)\mathrm{d}x=\int_a^b\rho(x)q(x)\omega_n(x)\mathrm{d}x+\int_a^b\rho(x)r(x)\mathrm{d}x.$$

如果对任何不超过 $n-1$ 次多项式 $q(x)$ 都有

$$\int_a^b\rho(x)q(x)\omega_n(x)\mathrm{d}x=0,\tag{4-22}$$

则求积公式(4-21) 对任何不超过 $2n-1$ 次的多项式精确成立. 也就是说，只要选取节点满足(4-22) 式，则求积公式的代数精确度就能达到 $2n-1$.

条件(4-22) 式表明 $\omega_n(x)$ 和 $q(x)$ 在 $[a,b]$ 上关于权函数 $\rho(x)$ 正交，从正交条件即可解出 $x_1,x_2,\cdots,x_n$. 实际上利用 $[a,b]$ 上关于权函数 $\rho(x)$ 的正交多项式 $\{P_n(x)\}$ 的性质，可以知道 $P_n(x)$ 的 n 个零点就是高斯型求积公式的 n 个节点，有了 n 个节点后就可以计算

$$A_k = \int_a^b \frac{\rho(x)\omega_n(x)}{(x-x_k)\omega'_n(x_k)}dx. \tag{4-23}$$

高斯型求积公式的误差有

定理7 设$f(x) \in C^{2n}[a,b]$,则高斯型求积公式的截断误差为

$$R(f) = \frac{f^{(2n)}(\xi)}{(2n)!}\int_a^b \rho(x)\omega_n^2(x)dx. \tag{4-24}$$

高斯型求积公式的优点是代数精确度高,但节点和系数的计算比较麻烦.由于正交多项式随权函数不同而不同,因此有各种不同类型的高斯型求积公式.

4.5.1 高斯-勒让德求积公式

高斯-勒让德(Gauss-Legendre)求积公式是古典的高斯型求积公式,它的积分区间是[-1,1],权函数$\rho(x)=1$,求积公式为

$$\int_{-1}^1 f(x)dx \approx \sum_{k=1}^n A_k f(x_k), \tag{4-25}$$

其中节点x_k是n次勒让德多项式的零点,勒让德多项式

$$L_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}[(x^2-1)^n],$$

与$L_0(x)=1$构成区间[-1,1]上的正交系,对一切$x^k(k=0,1,\cdots,n)$正交.公式(4-25)中系数A_k按(4-23)计算有

$$A_k = \frac{2(1-x_k^2)}{[n\,L_{n-1}(x_k)]^2}, \quad k=1,2,\cdots,n,$$

截断误差

$$R(f) = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3}f^{(2n)}(\xi), \quad \xi\in[-1,1].$$

表4-4给出了高斯-勒让德公式在$n=2,4,6,8$时的节点和系数.

表4-4

n	x_k	A_k	n	x_k	A_k
2	±0.5773503	1.0000000	8	±0.9602899	0.1012285
4	±0.8611363	0.3478548		±0.7966665	0.2223810
	±0.3399810	0.6521452		±0.5255324	0.3137066
6	±0.9324695	0.1713245		±0.1834346	0.3626838
	±0.6612094	0.3607616			
	±0.2386192	0.4679139			

4.5.2 高斯-拉盖尔求积公式

高斯-拉盖尔(Gauss-Laguerre)求积公式的积分区间是[0,∞),权函数$\rho(x)=e^{-x}$,求积公式为

$$\int_0^{\infty} e^{-x} f(x) dx \approx \sum_{k=1}^{n} A_k f(x_k), \tag{4-26}$$

其中节点 x_k 是 n 次拉盖尔多项式 $L_n(x)$ 的零点;拉盖尔多项式为

$$L_n(x) = e^x \frac{d^n}{dx^n}(e^{-x} x^n).$$

系数

$$A_k = \frac{(n!)^2}{x_k [L'_n(x_k)]^2},$$

截断误差

$$R(f) = \frac{(n!)^2}{(2n)!} f^{(2n)}(\xi).$$

节点 x_k 和系数 A_k 在 $n = 2,4,6,8$ 时,由表 4-4 给出.

表 4-4

n	x_k	A_k	n	x_k	A_k
2	0.5857864	0.8535534		9.8374674	0.0002610
	3.4142136	0.1464466		15.9828740	0.0000009
4	0.3225477	0.6031541	8	0.1702796	0.3691886
	1.7457611	0.3574187		0.9037018	0.4187868
	4.5366203	0.0388879		2.2510866	0.1757950
	9.3950709	0.0005393		4.2667002	0.0333435
6	0.2228466	0.4589647		7.0459054	0.0027945
	1.1889321	0.4170008		10.7585160	0.0000908
	2.9927363	0.1133734		15.7406786	0.0000008
	5.7751436	0.0103992		22.8631317	0.0000000

4.5.3 高斯 - 埃尔米特求积公式

高斯 - 埃尔米特求积公式的积分区间是$(-\infty, \infty)$,权函数 $\rho(x) = e^{-x^2}$,求积公式为

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{k=1}^{n} A_k f(x_k), \tag{4-27}$$

其中节点 x_k 是 n 次埃尔米特多项式 $H_n(x)$ 的零点;埃尔米特多项式为

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}).$$

系数

$$A_k = \frac{2^{n+1} n! \sqrt{\pi}}{[H'_n(x_k)]^2},$$

截断误差

$$R(f) = \frac{n! \sqrt{\pi}}{2^n (2n)!} f^{(2n)}(\xi).$$

高斯－埃尔米特求积公式 $n=2,4,6,8$ 时的节点 x_k 和系数 A_k 见表4-5.

表 4-5

n	x_k	A_k	n	x_k	A_k
2	±0.70710678	0.88622693	8	±0.38118699	0.66114701
4	±0.52464762	0.80491409		±1.15719371	0.20780233
	±1.65068012	0.08131284		±1.98165676	0.01707798
6	±0.43607741	0.72462960		±2.93063742	0.00019960
	±1.33584907	0.15706732			
	±2.35060497	0.00453001			

4.6 数值微分

当函数用表格给出时，导数就不可能通过极限来计算，只能用近似方法给出．下面给出一些常用的导数近似计算公式及相应的误差阶．令

$$y_0=f(x_i),\ y_k=f(x_{i+k}),\ y_{-k}=f(x_{i-k}),$$

$$x_i=x_0+ih,\quad i=0,1,\cdots,n,\ h=\frac{b-a}{n}.$$

1. 一阶导数

二点式：$f'(x_0)=\frac{1}{h}(y_1-y_0)+O(h)$;

$$f'(x_0)=\frac{1}{h}(y_0-y_{-1})+O(h).$$

三点式：$f'(x_0)=\frac{1}{2h}(-3y_0+4y_1-y_2)+O(h^2)$;

$$f'(x_0)=\frac{1}{2h}(y_1-y_{-1})+O(h^2);$$

$$f'(x_0)=\frac{1}{2h}(y_{-2}-4y_{-1}+3y_0)+O(h^2).$$

五点式：$f'(x_0)=\frac{1}{12h}(-25y_0+48y_1-36y_2+16y_3-3y_4)+O(h^4)$;

$$f'(x_0)=\frac{1}{12h}(-3y_{-1}-10y_0+18y_1-6y_2+y_3)+O(h^4);$$

$$f'(x_0)=\frac{1}{12h}(y_{-2}-8y_{-1}+8y_1-y_2)+O(h^4);$$

$$f'(x_0)=\frac{1}{12h}(-y_{-3}+6y_{-2}-18y_{-1}+10y_0+3y_1)+O(h^4);$$

$$f'(x_0)=\frac{1}{12h}(3y_{-4}-16y_{-3}+36y_{-2}-48y_{-1}+25y_0)+O(h^4).$$

2. 二阶导数

$$f''(x_0)=\frac{1}{h^2}(y_0-2y_1+y_2)+O(h);$$

$$f''(x_0)=\frac{1}{h^2}(y_{-1}-2y_0+y_1)+O(h^2);$$

$$f''(x_0)=\frac{1}{h^2}(y_{-2}-2y_{-1}+y_0)+O(h);$$

$$f''(x_0)=\frac{1}{h^2}(2y_0-5y_1+4y_2-y_3)+O(h^2);$$

$$f''(x_0)=\frac{1}{h^2}(-y_{-3}+4y_{-2}-5y_{-1}+2y_0)+O(h^2);$$

$$f''(x_0)=\frac{1}{12h^2}(35y_0-104y_1+114y_2-56y_3+11y_4)+O(h^4);$$

$$f''(x_0)=\frac{1}{12h^2}(11y_{-1}-20y_0+6y_1+4y_2-y_3)+O(h^4);$$

$$f''(x_0)=\frac{1}{12h^2}(-y_{-2}+16y_{-1}-30y_0+16y_1-y_2)+O(h^4);$$

$$f''(x_0)=\frac{1}{12h^2}(-y_{-3}+4y_{-2}+6y_{-1}-20y_0+11y_1)+O(h^4);$$

$$f''(x_0)=\frac{1}{12h^2}(11y_{-4}-56y_{-3}+114y_{-2}-104y_{-1}+35y_0)+O(h^4).$$

5　常微分方程初值问题的数值解法

一阶常微分方程初值问题的一般形式为

$$\begin{cases}\dfrac{dy}{dx}=f(x,y),\\ y(x_0)=y_0,\end{cases}\tag{5-1}$$

其中 f 是已知函数，$y(x_0)=y_0$ 是初始条件，y_0 已知.

在以后的讨论中，记问题(5-1)的准确解 y 在 x_n 点的值为 $y(x_n)$，用 y_n 表示近似解的值，$f_n=f(x_n,y_n)$ 表示 $f(x_n,y(x_n))$ 的近似值.

5.1　几个常用的定义

定义 1　若存在常数 $L>0$，使 $f(x,y)$ 在定义区域 D 上对所有 $(x,y_1),(x,y_2)\in D$ 有

$$|f(x,y_1)-f(x,y_2)|\leqslant L|y_1-y_2|,$$

则称 f 在 D 上对 y 满足**利普希茨(Lipschitz)条件**(简称利氏条件)，L 称为利普希茨常数(简称利氏常数).

求常微分方程初值问题(5-1)的数值解，是求 $y(x)$ 在存在区间 $[a,b]$ 中点列 $x_i=x_{i-1}+h_i,(i=1,2,\cdots)$ 的近似值 y_i，其中 h_i 称为步长. 若对所有 i 都有 $h_i=h$，称为等步长.

定义 2 在计算 y_{n+1} 时，只需用到 y_n、f_n 的方法称为**一步法**.

定义 3 在计算 y_{n+k} 时，需用到 y_{n+j}、$f_{n+j}(j=0,1,\cdots,k)$ 的线性组合的方法，即

$$\begin{aligned} y_{n+k} = {} & \alpha_0 y_n + \alpha_1 y_{n+1} + \cdots + \alpha_{k-1} y_{n+k-1} + \\ & h(\beta_0 f_n + \beta_1 f_{n+1} + \cdots + \beta_k f_{n+k}). \end{aligned} \tag{5-2}$$

其中 α_i、$\beta_i(i=0,1,\cdots,k)$ 是常数. α_0、β_0 不能全为零. 此方法称为**线性多步法**.

定义 4 在公式(5-2)中，若 $\beta_k=0$，则此方法称为**显式线性多步法**.

在公式(5-2)中，若 $k=1,\beta_1=0$，则此方法称为显式一步法. 显式一步法一般可表示为

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h). \tag{5-3}$$

定义 5 在公式(5-2)中，若 $\beta_k \neq 0$，则此方法称为**隐式线性多步法**. 若

$$y_{n+1} = y_n + h\varphi(x_n, y_n, y_{n+1}, h), \tag{5-4}$$

则称为隐式一步法.

定义 6 (收敛性) 若初值问题(5-1)的数值方法对任意固定的

$$x_n = x_0 + nh,$$

当 $h \to 0(n \to \infty)$ 时有 $y_n \to y(x_n)$，则称该数值方法收敛.

定义 7 假设前面各步都没有误差，即 $y(x_n) = y_n$ 准确成立，则称

$$T_{n+1} = y(x_{n+1}) - y_{n+1} \tag{5-5}$$

为**局部截断误差**，即计算一步产生的误差.

定义 8 用某种数值方法计算得到点 x_n 所对应的 y_n，称

$$e_n = y(x_n) - y_n \tag{5-6}$$

为**整体截断误差**，即整个计算过程的总误差.

常微分方程初值问题数值解的每一步计算都依赖前一步计算的结果，所以必须考虑前面误差对后面计算的影响，考虑误差的积累会不会盖过真解，这就是数值稳定性问题.

令

$$\rho_{n+1} = y_{n+1} - y_{n+1}^*, \tag{5-7}$$

其中 y_{n+1}^* 是 y_{n+1} 的近似值. 若 ρ_{n+1} 不大，y_{n+1}^* 可以作为 $y(x_{n+1})$ 的近似值.

讨论方法的稳定性常用**试验方程**，即方程

$$\begin{cases} \dfrac{dy}{dx} = \lambda y, \\ y(x_0) = y_0, \end{cases} \tag{5-8}$$

其中 λ 可以取复数，并设 $\mathrm{Re}\lambda > 0$. 这样可以保证微分方程(5-8)是稳定的.

用一步法解方程(5-8)可以得出

$$y_{n+1} = \nu(\lambda h) y_n. \tag{5-9}$$

在 y_0 上若有误差 ρ_0，并设以后的计算是精确的，则在 y_{n+1} 引起的误差为

$$\rho_{n+1} = \nu(\lambda h)\rho_n = (\nu(\lambda h))^{n+1}\rho_0.$$

定义 9 用一步法固定步长 h 解微分方程(5-1)，若在 y_n 上有误差 ρ，而由此引

起以后各步 $y_m(m > n)$ 的误差不超过 ρ,则称此一步法绝对稳定.也就是说,要求 $|\nu(\mu)| \leqslant 1$,其中 $\mu = \lambda h$.

将线性多步法(5-2)用到解初值问题(5-8),可得到一个 k 阶的常系数差分方程

$$\sum_{j=0}^{k}(\alpha_j - \mu\beta_j)y_{n+j} = 0,$$

它的特征方程是

$$\rho(\xi) - \mu\sigma(\xi) = 0, \tag{5-10}$$

其中 $\rho(\xi) = \sum_{j=0}^{k}\alpha_j\xi^j, \sigma(\xi) = \sum_{j=0}^{k}\beta_j\xi^j$. 令特征方程(5-10)的根为 $\xi_j(j = 1,2,\cdots,k)$.

定义 10 对于给定的 μ,若线性多步法所对应的特征方程(5-10)的根 ξ_j 满足 $|\xi_j| < 1(j = 1,2,\cdots,k)$,则称此线性多步法绝对稳定.

5.2 几种简单的一步法

5.2.1 欧拉方法

1. 欧拉(Euler)方法的计算公式

$$\begin{cases} y(x_0) = y_0, \\ y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0,1,2,\cdots. \end{cases} \tag{5-11}$$

2. 欧拉方法的几何意义

微分方程 $\frac{dy}{dx} = f(x,y)$ 在 x-y 平面的带形区域 $a \leqslant x \leqslant b, -\infty < y < +\infty$ 内确定了一个向量场,求解微分方程(5-1),从几何上看,是找一条通过点 (x_0,y_0) 的曲线 $y = y(x)$,使曲线上每一点的切线方向与已给向量场在该点的方向一致(图 5-1).欧拉计算公式(5-11)所定义的点 (x_n, y_n) 可以看作是通过准确初始点 (x_0,y_0) 的曲线多边形的顶点,而每条连线的方向由它左边终点的向量场决定.

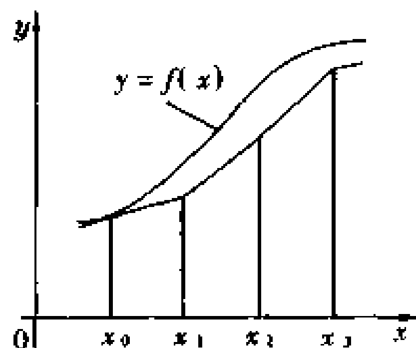


图 5-1

3. 欧拉公式的误差

令

$$\tilde{y}_{n+1} = y(x_n) + hf(x_n, y(x_n)),$$

则利用 $y(x_{n+1})$ 在 x_n 点的泰勒展开,容易得到欧拉公式的局部截断误差

$$T_{n+1} = y(x_{n+1}) - \tilde{y}_{n+1} = O(h^2), \tag{5-12}$$

整体截断误差

$$|e_{n+1}| = |y(x_{n+1}) - y_{n+1}| \leq |y(x_{n+1}) - \tilde{y}_{n+1}| + |\tilde{y}_{n+1} - y_{n+1}|.$$

根据 y_{n+1} 和 $\tilde{y}_{n+1}$ 的定义和 $f(x,y)$ 满足利氏条件,存在常数 L 使

$$|e_{n+1}| \leq |T_{n+1}| + (1 + hL)|e_n|.$$

这是第 $n+1$ 步与第 n 步的总体截断误差之间的关系,它对一切 n 成立.若注意到 $y(x_0) - y_0 = 0$,有总体截断误差 $|e_n| = O(h)$,这说明方法收敛.

4. 欧拉方法的稳定性

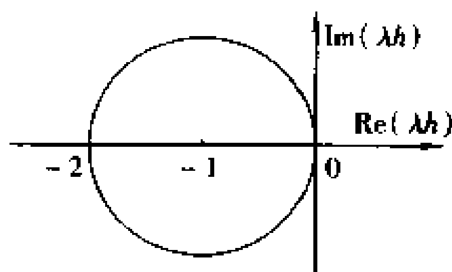


图 5-2

把欧拉方法用到方程(5-8)上,得到误差方程

$$\rho_{n+1} = \rho_n + \lambda h \rho_n.$$

要求误差不增长,即要求

$$\left|\frac{\rho_{n+1}}{\rho_n}\right| = |1 + \lambda h| \leq 1,$$

所以欧拉方法的绝对稳定区域是以 $h\lambda = -1$ 为中心、1 为半径的圆,如图 5-2.

欧拉方法是显式一步法.

5.2.2 隐式欧拉方法

1. 隐式欧拉方法的计算公式

$$\begin{cases} y(x_0) = y_0, \\ y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n = 0,1,2,\cdots. \end{cases} \tag{5-13}$$

2. 隐式欧拉公式的误差

隐式欧拉公式的局部截断误差

$$T_{n+1} = O(h^2).$$

因为方程(5-13)是隐式方程,计算时需要迭代,迭代格式为

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n), \\ y_{n+1}^{(k+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(k)}), \quad k = 0,1,2,\cdots. \end{cases} \tag{5-14}$$

利用利氏条件

$$|y_{n+1}^{(k+1)} - y_{n+1}^{(k)}| \leq hL|y_{n+1}^{(k)} - y_{n+1}^{(k-1)}|,$$

因此,当 $0 < hL < 1$ 时迭代过程收敛.总体截断误差当 $0 < hL < 1$ 时也是 $O(h)$.方法收敛.

3. 隐式欧拉方法的稳定性

隐式欧拉方法的误差方程为

$$\rho_{n+1} = \rho + \lambda h \rho_{n+1}.$$

因此,

$$\left|\frac{\rho_{n+1}}{\rho_n}\right| = \frac{1}{|1 - \lambda h|} = \frac{1}{\sqrt{1 - 2\mathrm{Re}(\lambda h) + h^2|\lambda|^2}}.$$

只要 $\mathrm{Re}(\lambda h) < 0$，就有 $|\rho_{n+1}/\rho_n| < 1$，隐式欧拉方法的绝对稳定区域是 λh- 复平面的整个左半平面. 称这种情形是 A 稳定的.

5.2.3 梯形方法

1. 梯形方法的计算公式

$$\begin{cases} y(x_0) = y_0, \\ y_{n+1} = y_n + \dfrac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_{n+1})). \end{cases} \tag{5-15}$$

梯形方法也是隐式方法，所以计算时要迭代，迭代格式为

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n), \\ y_{n+1}^{(k+1)} = y_n + \dfrac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})), \quad k = 0,1,2,\cdots. \end{cases} \tag{5-16}$$

因为

$$|y_{n+1}^{(k+1)} - y_n^{(k)}| \leqslant \frac{hL}{2}|y_{n+1}^{(k)} - y_{n+1}^{(k-1)}|,$$

因此，当 $0 < \dfrac{hL}{2} < 1$ 时迭代收敛.

2. 梯形公式的误差

梯形公式的局部截断误差为 $O(h^3)$. 当 $0 < \dfrac{hL}{2} < 1$ 时迭代收敛，总体截断误差为 $O(h^2)$.

3. 梯形方法的稳定性

梯形公式的误差方程为

$$\rho_{n+1} = \rho_n + \frac{\lambda h}{2}(\rho_n + \rho_{n+1}).$$

当 $\mathrm{Re}(\lambda h) < 0$ 时，$|\rho_{n+1}/\rho_n| \leqslant 1$，所以梯形公式是 A 稳定的.

5.3 龙格 - 库塔方法

龙格 - 库塔(Kutla)方法简称为 **R-K 方法**，它是用计算不同点上的函数值，并对这些函数值作线性组合，构成公式，再把近似公式的解和泰勒展开比较，给出公式的误差. R-K 方法是显式一步方法.

5.3.1 二阶 R-K 方法

1. 改进的欧拉方法

$$y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2). \tag{5-17}$$

其中

$$K_1 = f(x_n, y_n),$$
$$K_2 = f(x_n + h, y_n + hK_1).$$

2. 休恩(Heun)二阶方法

$$y_{n+1} = y_n + \frac{h}{4}(K_1 + 3K_2), \tag{5-18}$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hK_1\right).$$

5.3.2　三阶 R-K 方法

1. 休恩三阶方法

$$y_{n+1} = y_n + \frac{h}{4}(K_1 + 3K_3), \tag{5-19}$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f\left(x_n + \frac{h}{3}, y_n + \frac{h}{3}K_1\right),$$

$$K_3 = f\left(x_n + \frac{2h}{3}, y_n + \frac{2}{3}hK_2\right).$$

2. 库塔方法

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 4K_2 + K_3), \tag{5-20}$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right),$$

$$K_3 = f(x_n + h, y_n - hK_1 + 2hK_2).$$

5.3.3　四阶 R-K 方法

1. 经典 R-K 方法

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4). \tag{5-21}$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{h}{2}K_1\right),$$

$$K_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{h}{2}K_2\right),$$

$$K_4 = f(x_n + h, y_n + hK_3).$$

经典 R-K 方法的稳定区域为

$$\left|1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{\mu^4}{24}\right| \leqslant 1,$$

即在 μ 平面上由曲线

$$1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{\mu^4}{24} = e^{i\theta}$$

围成的区域，其中 $\mu = \lambda h$，见图 5-3.

2. 基尔(Gill) 方法

$$y_{n+1} = y_n + \frac{h}{8}(K_1 + 3K_2 + 3K_3 + K_4), \tag{5-22}$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f(x_n + \frac{1}{3}h, y_n + \frac{h}{3}K_1),$$

$$K_3 = f(x_n + \frac{2}{3}h, y_n - \frac{h}{3}(K_1 - 3K_2)),$$

$$K_4 = f(x_n + h, y_n + h(K_1 - K_2 + K_3)).$$

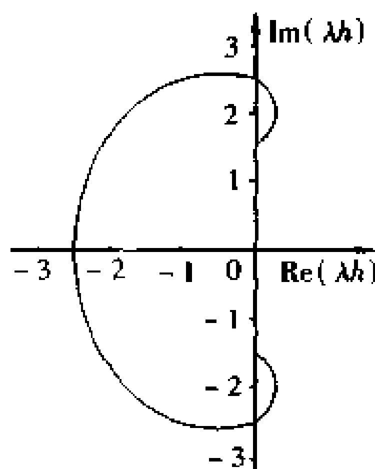


图 5-3

5.3.4 高阶 R-K 方法

库塔 - 尼斯特龙(Nyström) 五阶方法

$$y_{n+1} = y_n + \frac{h}{192}(23K_1 + 125K_3 - 81K_5 + 125K_6), \tag{5-23}$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f(x_n + \frac{1}{3}h, y_n + \frac{h}{3}K_1),$$

$$K_3 = f(x_n + \frac{2}{5}h, y_n + \frac{h}{25}(4K_1 + 6K_2)),$$

$$K_4 = f(x_n + h, y_n + \frac{h}{4}(K_1 - 12K_2 + 15K_3)),$$

$$K_5 = f(x_n + \frac{2}{3}h, y_n + \frac{h}{81}(6K_1 + 90K_2 - 50K_3 + 8K_4)),$$

$$K_6 = f(x_n + \frac{4}{5}h, y_n + \frac{h}{75}(6K_1 + 36K_2 + 10K_3 + 8K_4)).$$

5.4 线性多步法

线性多步法是将积分恒等式

$$y(x_{n+k}) = y(x_n) + \int_{x_n}^{x_{n+k}} f(t, y(t))\mathrm{d}t$$

中的被积函数 $f(t, y(t))$ 用各种不同的插值多项式来近似表达而得到的方法.

5.4.1 亚当斯显式线性多步法

将积分恒等式

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(t, y(t))\mathrm{d}t \tag{5-24}$$

中的被积函数 $f(t,y(t))$ 用 k 个节点 $x_n,x_{n-1},\cdots,x_{n-k+1}$ 及相应的函数值 $f_n,f_{n-1},\cdots,f_{n-k+1}$ 所作的 $k-1$ 次插值多项式代替,即

$$f(t,y(t))\approx l_n(t)f_n+l_{n-1}(t)f_{n-1}+\cdots+l_{n-k+1}f_{n-k+1},$$

其中 $l_i(t)$ 是对应节点上的插值基函数,由(2-3)式给出,就得到**亚当斯(Adams)显式线性多步法公式**

$$y_{n+1}=y_n+h(a_0f_n+a_1f_{n-1}+\cdots+a_{k-1}f_{n-k+1}),$$

其中

$$a_j=\frac{1}{h}\int_{x_n}^{x_{n+1}}l_{n-j}(t)\mathrm{d}t,\quad j=0,1,\cdots,k-1.$$

k 从1-5的各项系数和局部截断误差如表5-1所示.

表 5-1

k	a_0	a_1	a_2	a_3	a_4	局部截断误差
1	1					$\frac{1}{2}h^2y''(\xi)$
2	$\frac{3}{2}$	$-\frac{1}{2}$				$\frac{5}{12}h^3y^{(3)}(\xi)$
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$			$\frac{3}{8}h^4y^{(4)}(\xi)$
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$		$\frac{251}{720}h^5y^{(5)}(\xi)$
5	$\frac{1901}{720}$	$-\frac{2774}{720}$	$\frac{2616}{720}$	$-\frac{1274}{720}$	$\frac{251}{720}$	$\frac{95}{288}h^6y^{(6)}(\xi)$

注:ξ 属于插值区间.

最常用的亚当斯四阶显式线性多步法公式为

$$y_{n+1}=y_n+\frac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}).\tag{5-25}$$

亚当斯四阶显式线性多步法公式的误差方程是

$$\rho_{n+1}=\rho_n+\frac{\lambda h}{24}(55\rho_n-59\rho_{n-1}+37\rho_{n-2}-9\rho_{n-3}),$$

相应的特征方程是

$$\xi^4-\xi^3=\frac{\lambda h}{24}(55\xi^3-59\xi^2+37\xi-9),$$

稳定条件是

$$|\xi_j(\lambda h)|<1,\quad j=1,2,3,4,$$

稳定区域如图5-4所示.

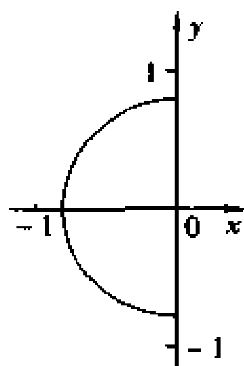


图 5-4

5.4.2 亚当斯隐式线性多步法

对积分恒等式(5-24)中的被积函数，用节点 $x_{n+1},x_n,\cdots,x_{n-k+2}$ 和相应的函数值 $f_{n+1},f_n,\cdots,f_{n-k+2}$ 作 $k-1$ 次插值多项式，则可以得到亚当斯隐式线性多步法公式

$$y_{n+1}=y_n+h(a_0f_{n+1}+a_1f_n+\cdots+a_{k-1}f_{n-k+2}),$$

其中

$$a_j=\frac{1}{h}\int_{x_n}^{x_{n+1}}l_{n+1-j}(t)\mathrm{d}t,\quad j=0,1,\cdots,k-1,$$

$l_{n+1-j}(t)$ 是节点 x_{n+1-j} 上的插值基函数.

k 从 1-5 的各项系数和局部截断误差如表 5-2 所示.

表 5-2

k	a_0	a_1	a_2	a_3	a_4	局部截断误差
1	1					$-\frac{1}{2}h^2y''(\xi)$
2	$\frac{1}{2}$	$\frac{1}{2}$				$-\frac{1}{12}h^3y^{(3)}(\xi)$
3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$			$-\frac{1}{24}h^4y^{(4)}(\xi)$
4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$		$-\frac{19}{720}h^5y^{(5)}(\xi)$
5	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$	$-\frac{3}{160}h^6y^{(6)}(\xi)$

最常用的亚当斯四阶隐式线性多步法公式为

$$y_{n+1}=y_n+\frac{h}{24}(9f_{n+1}+19f_n-5f_{n-1}+f_{n-2}). \tag{5-26}$$

用隐式公式计算需要迭代，迭代格式为

$$\begin{cases}y_{n+1}^{(0)}=y_n+\dfrac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}),\\ y_{n+1}^{(k+1)}=y_n+\dfrac{h}{24}(9f(x_{n+1},y_{n+1}^{(k)})+19f_n-5f_{n-1}+f_{n-2}),\quad k=0,1,2,\cdots\end{cases}$$

也就是用(5-25)式计算初值，再用(5-26)式进行迭代. 迭代收敛的条件为

$$0<\frac{3}{8}hL<1,$$

L 是利氏常数.

对应于(5-26)式的特征方程是

$$\xi^3-\xi^2=\frac{\lambda h}{24}(9\xi^3+19\xi^2-5\xi+1).$$

稳定条件为

$$|\xi_j(\lambda h)|\leq 1,\quad j=1,2,3,$$

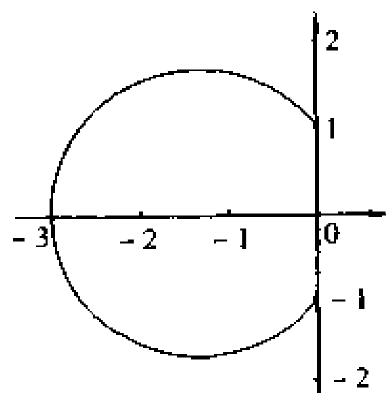


图 5-5

稳定区域如图 5-5 所示.

5.4.3　其它几种四阶多步法

1. 米尔恩方法

米尔恩(Milne)方法是四阶显式多步法,计算公式为

$$y_{n+1} = y_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}). \tag{5-27}$$

米尔恩方法的局部截断误差为$\frac{14}{45}h^5y^{(5)}(\xi)$.

2. 辛普森方法

辛普森方法是四阶隐式方法,计算公式为

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}). \tag{5-28}$$

迭代时的初值可选用任一种四阶显式格式计算.其局部截断误差为 $-\frac{1}{90}h^5y^{(5)}(\xi)$.

3. 汉明方法

汉明(Hamming)方法是四阶隐式方法,计算公式为

$$y_{n+1} = \frac{1}{8}(9y_n - y_{n-2}) + \frac{3h}{8}(f_{n+1} + 2f_n - f_{n-1}), \tag{5-29}$$

迭代初值选用四阶显式格式计算.其局部截断误差为 $-\frac{h^5}{40}y^{(5)}(\xi)$.

5.5　预估-校正方法

用显式方法计算初值,用隐式方法进行迭代,实际上就是一个预估-校正过程,只是迭代的次数由给定精度控制.这里介绍的**预估-校正方法**不进行迭代,是事先规定校正的方法.当然,一般在选择预估公式和校正公式时,误差的阶应该匹配.

1. 改进的欧拉预估-校正方法

预估　$y_{n+1}^{(0)} = y_n + hf(x_n, y_n)$,

计算　$m_{n+1} = f(x_{n+1}, y_{n+1}^{(0)})$,

校正　$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + m_{n+1})$.

2. 亚当斯四阶预估-校正方法

预估　$y_{n+1}^{(0)} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$,

计算　$m_{n+1} = f(x_{n+1}, y_{n+1}^{(0)})$,

校正 $y_{n+1} = y_n + \frac{h}{24}(9m_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$.

3. 改进的亚当斯四阶预估 - 校正方法

用 p_n 和 c_n 分别表示第 n 步 y_n 的预估值和校正值，再利用四阶亚当斯公式的局部截断误差对计算值进行修正，给出新的预估 - 校正方法.

预估 $p_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$,

改进 $m_{n+1} = p_{n+1} + \frac{251}{270}(c_n - p_n)$,

计算 $m'_{n+1} = f(x_{n+1}, m_{n+1})$,

校正 $c_{n+1} = y_n + \frac{h}{24}(9m'_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$,

改进 $y_{n+1} = c_{n+1} + \frac{19}{270}(p_{n+1} - c_{n+1})$,

计算 $f_{n+1} = f(x_{n+1}, y_{n+1})$.

开始的 c_n、p_n 可令其为零.

4. 汉明预估 - 校正方法

汉明预估 - 校正方法是把米尔恩公式(5-27)和汉明公式(5-29)相配合，利用局部截断误差作改进而得到的.

预估 $p_{n+1} = y_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2})$,

改进 $m_{n+1} = p_{n+1} - \frac{112}{121}(p_n - c_n)$,

计算 $m'_{n+1} = f(x_{n+1}, m_{n+1})$,

校正 $c_{n+1} = \frac{9}{8}y_n - \frac{1}{8}y_{n-2} + \frac{3h}{8}(m'_{n+1} + 2f_n - f_{n-1})$,

改进 $y_{n+1} = c_{n+1} + \frac{9}{121}(p_{n+1} - c_{n+1})$,

计算 $f_{n+1} = f(x_{n+1}, y_{n+1})$.

5.6 常微分方程组和高阶方程初值问题的数值解

5.6.1 一阶常微分方程组初值问题和高阶方程初值问题的一般形式

1. 一阶方程组初值问题的一般形式

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, y_2, \cdots, y_n), & y_1(x_0) = y_{10}, \\ \frac{dy_2}{dx} = f_2(x, y_1, y_2, \cdots, y_n), & y_2(x_0) = y_{20}, \\ \vdots \\ \frac{dy_m}{dx} = f_m(x, y_1, y_2, \cdots, y_n), & y_m(x_0) = y_{m0}. \end{cases} \tag{5-30}$$

2. 高阶方程初值问题的一般形式

$$\begin{cases} \dfrac{d^n y}{dx^n} = f\left(x, y, \dfrac{dy}{dx}, \cdots, \dfrac{d^{n-1}y}{dx^{n-1}}\right), \\ y(x_0) = y_0, \dfrac{dy(x_0)}{dx} = y_0^{(1)}, \cdots, \dfrac{d^{n-1}y(x_0)}{dx^{n-1}} = y_0^{(n-1)}. \end{cases} \tag{5-31}$$

如果设

$$y_1 = y, \quad y_2 = \frac{dy}{dx}, \cdots, \quad y_n = \frac{d^{n-1}y}{dx^{n-1}},$$

则高阶方程可化成方程组的形式：

$$\begin{cases} \dfrac{dy_1}{dx} = y_1, & y_1(x_0) = y_0, \\ \vdots & \vdots \\ \dfrac{dy_{n-1}}{dx} = y_n, & y_{n-1}(x_0) = y_0^{(n-2)}, \\ \dfrac{dy_n}{dx} = f(x, y_1, y_2, \cdots, y_n), & y_n(x_0) = y_0^{(n-1)}. \end{cases} \tag{5-32}$$

5.6.2　一阶方程组和高阶方程的数值解

前面介绍的一阶常微分方程的各种数值解法对方程组和高阶方程同样适用，我们只以几个最常用的公式为例给出计算公式.

1. 欧拉公式

$$\begin{cases} y_{l,n+1} = y_{ln} + hf_l(x_n, y_{1n}, y_{2n}, \cdots, y_{mn}), \\ y_l(x_0) = y_{l0}, \quad l = 1,2,\cdots,m. \end{cases} \tag{5-33}$$

2. 经典 R-K 公式

$$y_{l,n+1} = y_{ln} + \frac{h}{6}(K_{1l} + 2K_{2l} + 2K_{3l} + K_{4l}), \tag{5-34}$$

其中

$$\begin{aligned} K_{1l} &= f_l(x_n, y_{1n}, y_{2n}, \cdots, y_{mn}), \\ K_{2l} &= f_l\left(x_n + \frac{h}{2}, y_{1n} + \frac{h}{2}K_{11}, y_{2n} + \frac{h}{2}K_{12}, \cdots, y_{mn} + \frac{h}{2}K_{1m}\right), \\ K_{3l} &= f_l\left(x_n + \frac{h}{2}, y_{1n} + \frac{h}{2}K_{21}, y_{2n} + \frac{h}{2}K_{22}, \cdots, y_{mn} + \frac{h}{2}K_{2m}\right), \\ K_{4l} &= f_l(x_n + h, y_{1n} + hK_{31}, y_{2n} + hK_{32}, \cdots, y_{mn} + hK_{3m}), \\ & \qquad l = 1,2,\cdots,m. \end{aligned}$$

3. 亚当斯公式

(1) 外插

$$y_{l,n+1} = y_{ln} + \frac{h}{24}(55f_{ln} - 59f_{l,n-1} + 37f_{l,n-2} - 9f_{l,n-3}), \quad l = 1,2,\cdots,m. \tag{5-35}$$

(2) 内插

$$y_{l,n+1} = y_{ln} + \frac{h}{24}(9f_{l,n+1} + 19f_{ln} - 5f_{l,n-1} + f_{l,n-2}), \quad l = 1,2,\cdots,m. \tag{5-36}$$

4. 亚当斯预估 - 校正公式

$$\begin{cases} p_{l,n+1} = y_{ln} + \frac{h}{24}(55f_{ln} - 59f_{l,n-1} + 37f_{l,n-2} - 9f_{l,n-3}), \\ m_{l,n+1} = p_{l,n+1} + \frac{251}{270}(c_{ln} - p_{ln}), \\ m'_{l,n+1} = f_l(x_{n+1}, m_{1,n+1}, m_{2,n+1}, \cdots, m_{m,n+1}), \\ c_{l,n+1} = y_{ln} + \frac{h}{24}(9m'_{l,n+1} + 19f_{ln} - 5f_{l,n-1} + f_{l,n-2}), \\ y_{l,n+1} = c_{l,n+1} + \frac{19}{270}(p_{l,n+1} - c_{l,n+1}), \\ f_{l,n+1} = f_l(x_{n+1}, y_{1,n+1}, \cdots, y_{m,n+1}), \end{cases} \quad l = 1,2,\cdots,m. \tag{5-37}$$

5.7 刚性方程组的数值解法

5.7.1 刚性方程组的定义

定义 11 考虑 m 阶线性常系数方程组,并用向量形式表示为

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{J}\boldsymbol{y} + \boldsymbol{f}(x), \tag{5-38}$$

其中 $\boldsymbol{J}$ 是 $m \times m$ 阶常数矩阵.若它的特征值 $\lambda_j (j = 1,2,\cdots,m)$ 满足条件

1° $\mathrm{Re}\lambda_j < 0 \quad (j = 1,2,\cdots,m)$;

2° $s = \dfrac{\max\limits_j |\mathrm{Re}\lambda_j|}{\min\limits_j |\mathrm{Re}\lambda_j|} \gg 1$.

则称方程组(5-38)为**刚性方程组**,s 称为**刚性比**.

用数值方法计算刚性方程组的困难在于特征值相差悬殊,而解方程组时步长 h 的选取要对所有特征值 λ、λh 均属于稳定区域,所以计算刚性方程组的数值方法都是 A 稳定的.

5.7.2 吉尔方法

吉尔(Gear)方法是如下的隐式多步法:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\beta_k y_{n+k}, \tag{5-39}$$

其中系数 α_j、β_k 的选取如表 5-3 所示.

表 5-3

k	β_k	α_6	α_5	α_4	α_3	α_2	α_1	α_0
1	1						1	-1
2	$\frac{2}{3}$					1	$-\frac{3}{4}$	$\frac{1}{3}$
3	$\frac{6}{11}$				1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$
4	$\frac{12}{25}$			1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$
5	$\frac{60}{137}$		1	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$
6	$\frac{60}{147}$	1	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$

6　常微分方程边值问题的数值解

6.1　常微分方程边值问题

常微分方程边值问题的典型例子是二阶微分方程两点边值问题,一般表示为

$$y'' = f(x,y,y'),\quad a \leqslant x \leqslant b. \tag{6-1}$$

边界条件有三类:

第Ⅰ类　$y(a) = \alpha, y(b) = \beta$.

第Ⅱ类　$y'(a) = \alpha, y'(b) = \beta$.

第Ⅲ类　$y(a) - \alpha_1 y'(a) = \alpha, y(b) + \beta_1 y'(b) = \beta$,其中 $\alpha_1, \beta_1 \geqslant 0, \alpha_1 + \beta_1 > 0$.

若方程(6-1)是线性的,第Ⅰ类边值问题可表示为

$$\begin{cases} y'' = p(x)y' + q(x)y + r(x), a \leqslant x \leqslant b, \\ y(a) = \alpha, y(b) = \beta. \end{cases} \tag{6-2}$$

问题(6-2)中的系数若满足条件:

1° $p(x), q(x), r(x)$ 在 $[a,b]$ 上连续;

2° $q(x) > 0$ 对任意 $x \in [a,b]$,

则问题(6-2)的解存在而且唯一.

现在考虑两个初值问题:

$$\begin{cases} y''_1 = p(x)y'_1 + q(x)y_1 + r(x),\quad a \leqslant x \leqslant b, \\ y_1(a) = \alpha, y'_1(a) = 0 \end{cases} \tag{6-3}$$

及

$$\begin{cases} y''_2 = p(x)y'_2 + q(x)y_2,\quad a \leqslant x \leqslant b, \\ y_2(a) = 0, y'_2(a) = 1. \end{cases} \tag{6-4}$$

定理1 设 y_1 是初值问题(6-3)的解，y_2 是初值问题(6-4)的解，并设 $y_2(b)\neq 0$，则问题(6-2)的解可表示为问题(6-3)和(6-4)解的线性组合，即

$$y(x)=y_1(x)+\frac{\beta-y_1(b)}{y_2(b)}y_2(x). \tag{6-5}$$

6.2 打 靶 法

6.2.1 线性边值问题的打靶法

根据6.1，二阶线性边值问题可以化为两个初值问题解的线性组合，因此，可以利用解常微分方程初值问题的方法求解(6-2).

定理2 设问题(6-3)和(6-4)的近似解在 x_n 点的近似值分别为 y_{1n} 和 y_{2n}，它们的误差是 $O(h^p)$，则问题(6-2)在 x_n 处的近似值 y_n 的误差亦为 $O(h^p)$.

有时我们也可以将问题(6-2)的解看成是

$$\begin{cases} y''_1=p(x)y'_1+q(x)y_1+r(x), & a\leqslant x\leqslant b,\\ y_1(b)=\beta, y'_1(b)=0 \end{cases}$$

及

$$\begin{cases} y''_2=p(x)y'_2+q(x)y_2, & a\leqslant x\leqslant b,\\ y_2(b)=0, y'_2(b)=1 \end{cases}$$

两个初值问题的线性组合，这时问题(6-2)的解可表示为

$$y(x)=y_1(x)+\frac{\alpha-y_1(a)}{y_2(a)}y_2(x), \quad y_2(a)\neq 0.$$

6.2.2 非线性边值问题的打靶法

对方程

$$\begin{cases} y''=f(x,y,y'), & a\leqslant x\leqslant b,\\ y(a)=\alpha, y(b)=\beta \end{cases} \tag{6-6}$$

引进参数 t_k，初值问题

$$\begin{cases} y''=f(x,y,y'), & a\leqslant x\leqslant b,\\ y(a)=\alpha, y'(a)=t_k \end{cases} \tag{6-7}$$

的解与 t_k 有关，记为 $y(x,t_k)$. 用初值问题数值解的算法求出问题(6-7)的解 $y(x,t_k)$，若

$$|y(b,t_k)-\beta|<\varepsilon,$$

ε 是允许误差，则称 $y(x,t_k)$ 是边值问题(6-6)的近似解. 否则，将 t_k 修改为 t_{k+1} 后再解问题(6-7).

也就是说，对一系列 $\{t_k\}$ 进行试算，并要求 $\lim\limits_{k\to\infty}y(b,t_k)=\beta$. 因此，打靶法的关键是如何选择序列 $\{t_k\}$. 这可以用解非线性方程的一些办法来实现.

例如，用二分法来选择 t_k. 记

$$F(t_k) = y(b, t_k) - \beta.$$

在打靶法中选择 t_k 和t_{k+1}时要求 $F(t_k) \cdot F(t_{k+1}) < 0$,这时$[t_k, t_{k+1}]$就是迭代区间.

令

$$t_{k+2} = \frac{1}{2}(t_k + t_{k+1}),$$

即将区间$[t_k, t_{k+1}]$对半分.若 $F(t_{k+2}) \cdot F(t_k) < 0$,则用$[t_k, t_{k+2}]$代替$[t_k, t_{k+1}]$,否则用$[t_{k+2}, t_{k+1}]$代替$[t_k, t_{k+1}]$后继续迭代.

6.3 边值问题的差分解法

将区间$[a, b]$ n 等分:

$$x_i = a + ih, \quad i = 0, 1, \cdots, m,$$

$$h = \frac{b - a}{n},$$

利用数值微分的公式,用差商来代替微分方程中的导数.

在选用数值微分公式时,要注意截断误差的匹配,例如对一阶和二阶导数都选用截断误差为 $O(h^2)$ 的差商:

$$\begin{cases} y''(x_i) = \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2), \\ y'(x_i) = \dfrac{y_{i+1} - y_{i-1}}{2h} + O(h^2). \end{cases} \tag{6-8}$$

6.3.1 线性边值问题的差分方法

对方程(6-2)中的导数用差商(6-8)代替,并略去误差项 $O(h^2)$,则可以得到第Ⅰ边值问题的差分方程组

$$\begin{cases} y_0 = \alpha, \\ \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - p(x_i)\dfrac{y_{i+1} - y_{i-1}}{2h} - q(x_i)y_i = r(x_i), \\ \qquad\qquad\qquad\qquad\qquad\qquad i = 1, 2, \cdots, n-1, \\ y_n = \beta. \end{cases} \tag{6-9}$$

对第Ⅱ类、第Ⅲ类边值问题,边界点的差分近似也应保持相应的截断误差,即也取截断误差阶为 $O(h^2)$:

$$y'(x_0) = \frac{-y_2 + 4y_1 - 3y_0}{2h} + O(h^2),$$

$$y'(x_n) = \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} + O(h^2).$$

这样第Ⅱ类边值问题的差分方程组为

$$
\begin{cases}
\dfrac{-y_2+4y_1-3y_0}{2h}=\alpha, \\
\dfrac{y_{i+1}-2y_i+y_{i-1}}{h^2}-p(x_i)\dfrac{y_{i+1}-y_{i-1}}{2h}-q(x_i)y_i=r(x_i), \\
\qquad\qquad i=1,2,\cdots,n-1, \\
\dfrac{3y_n-4y_{n-1}+y_{n-2}}{2h}=\beta.
\end{cases}
\tag{6-10}
$$

第Ⅲ类边值问题的差分方程组为

$$
\begin{cases}
\dfrac{-y_2+4y_1-3y_0}{2h}-\alpha_1y_0=\alpha, \\
\dfrac{y_{i+1}-2y_i+y_{i-1}}{h^2}-p(x_i)\dfrac{y_{i+1}-y_{i-1}}{2h}-q(x_i)y_i=r(x_i), \\
\qquad\qquad i=1,2,\cdots,n-1, \\
\dfrac{3y_n-4y_{n-1}+y_{n-2}}{2h}+\beta_1y_n=\beta.
\end{cases}
\tag{6-11}
$$

因为线性微分方程可以简化为

$$\frac{\mathrm{d}y^2}{\mathrm{d}t^2}+Q(t)y=R(t),$$

因此,可以考虑在问题(6-2)中 $p(x)\equiv 0$ 的情形.

定理3 设在$[a,b]$上 $q(x)\geqslant 0, p(x)\equiv 0$,边值问题(6-2)的解为 $y(x)$,$y\in C^4[a,b]$,$M_4=\max\limits_{a\leqslant x\leqslant b}|f^{(4)}(x)|$ 及 $y_i(i=0,1,\cdots,n)$ 为方程(6-9)的解,则

$$|y(x_i)-y_i|\leqslant\frac{M_4(b-a)^2}{96}h^2.$$

定理3说明,若 $x=x_i$ 固定,当 $h\to 0$ 时,差分方程的解收敛到微分方程的解.

差分方程组(6-9)~(6-11)是线性代数方程组,对它们的求解可参阅数值代数中介绍的算法.

6.3.2 非线性边值问题的差分方法

对非线性方程边值问题(6-6)可类似地建立差分方程组

$$
\begin{cases}
y_0=\alpha, \\
\dfrac{y_{i+1}-2y_i+y_{i-1}}{h^2}-f\left(x_i,y_i,\dfrac{y_{i+1}-y_{i-1}}{2h}\right)=0, \\
\qquad\qquad i=1,2,\cdots,n-1, \\
y_n=\beta.
\end{cases}
\tag{6-12}
$$

定理4 若微分方程边值问题(6-6)式的解存在而且唯一,并且

$$L=\max_{(x,y,y')\in D}\left|\frac{\partial f(x,y,y')}{\partial y'}\right|,$$

其中 $D=\{(x,y,y')\mid a\leqslant x\leqslant b,-\infty<y<+\infty,-\infty<y'<+\infty\}$.如果 $h\leqslant 2/L$,则方程(6-12)有唯一解.

解方程(6-12)的数值方法可选用数值代数中解非线性方程组的牛顿法.

7　椭圆型偏微分方程的差分解法

7.1　椭圆型方程及定解条件

7.1.1　方程

常见的椭圆型偏微分方程为

$$\mathrm{L}u=-\left(\frac{\partial}{\partial x}\left(\beta\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\beta\frac{\partial u}{\partial y}\right)\right)=f, \tag{7-1}$$

其中 L 是微分算子, β、f 是给定的 x、y 的连续函数, $\beta>0$.

若方程(7-1)中的 $\beta\equiv1$, $f\equiv0$,则它就是拉普拉斯方程:

$$\mathrm{L}u=-\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y_2}\right)=0. \tag{7-2}$$

若方程(7-1)中的 $\beta\equiv1$, $f\not\equiv0$,则它就是泊松方程:

$$\mathrm{L}u=-\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=f. \tag{7-3}$$

7.1.2　定解条件

椭圆型方程的定解条件是边值条件,一般边界条件的给法有:

第 Ⅰ 类边界条件　在 ∂G 上给定函数值,

$$u|_{\partial G}=\overline{u}(x,y).$$

第 Ⅱ 类边界条件　在 ∂G 上给定外法向导数,

$$\left.\frac{\partial u}{\partial n}\right|_{\partial G}=p(x,y).$$

第 Ⅲ 类边界条件　在 ∂G 上给定函数及其外法向导数的线性组合,

$$\left.\left(\frac{\partial u}{\partial n}+\eta u\right)\right|_{\partial G}=q(x,y),$$

其中 η、q 都是给定的 x、y 的函数, $\eta\geqslant0$.

7.2　网格剖分和差分近似

7.2.1　网格剖分

设 G 为 x-y 平面上的一有界区域, ∂G 为其边界,如图 7-1.将 x-y 平面用两组平行于坐标轴的直线分割:

$$x = x_i = ih, \quad y = y_j = jk \quad i,j = 0, \pm 1, \pm 2, \cdots$$

h 和 k 称为**网格步长**，平行直线的交点称为**节点**，直线 ih 和 jk 的交点记为 (i,j). 属于 $G + \partial G$ 的节点称为**内点**；如果两个节点沿 x 方向（或沿 y 方向）只相差一个步长，称两个节点为相邻节点；若一个节点的所有四个相邻节点均属于 $G + \partial G$，则称该节点为**正则内点**，四个相邻节点中至少有一个不属于 $G + \partial G$，则称为**非正则内点**. 网格线与边界 ∂G 的交点称为**边界点**.

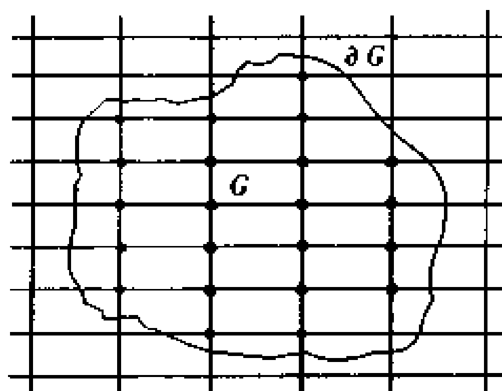


图 7-1

内点的全体构成网格区域，用 G_h 表示，并把全部正则内点的集合（图 7-1 中画 0 者）记为 $G_h^{(1)}$，全部非正则内点的集合（图 7-1 中画 · 者）记为 $G_h^{(2)}$，则 $G_h = G_h^{(1)} + G_h^{(2)}$. 边界点全体记为 ∂G_h.

7.2.2　正则内点的差分近似

1. 五点差分格式

令 $u(x_i, y_j) = u_{i,j}$, $\beta(x_i, y_j) = \beta_{i,j}$, $f(x_i, y_j) = f_{i,j}$，把数值微分公式

$$\left(\frac{\partial u}{\partial x}\right)_{(i,j)} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2),$$

$$\left(\frac{\partial u}{\partial y}\right)_{(i,j)} = \frac{u_{i,j+1} - u_{i,j-1}}{2k} + O(k^2),$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(i,j)} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + O(h^2),$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{(i,j)} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{k^2} + O(k^2)$$

代入(7-1)式，并略去误差项 $O(h^2 + k^2)$，则得到点 (i,j) 的差分格式

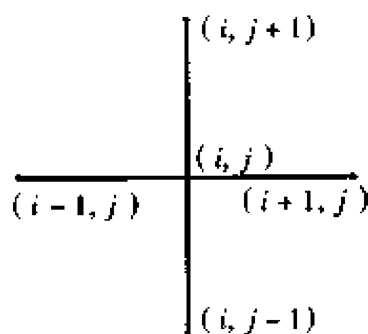


图 7-2

$$\Delta_{h,k} u_{i,j} = -\left[\frac{\beta_{i,j}}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1}{4h^2}(\beta_{i+1,j} - \beta_{i-1,j})(u_{i+1,j} - u_{i-1,j}) + \frac{\beta_{i,j}}{k^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) + \frac{1}{4k^2}(\beta_{i,j+1} - \beta_{i,j-1})(u_{i,j+1} - u_{i,j-1})\right] = f_{i,j}, \tag{7-4}$$

其中 $\Delta_{h,k}$ 是差分算子，下标 h 和 k 是网格步长. 差分格式用到的网格结点如图 7-2 所示. 因用到的是(i,j)自身及相邻的四个节点，所以叫做**五点差分格式**. 用五点差分格式逼近(7-1)式，其截断误差为 $O(h^2+k^2)$.

对拉普拉斯方程则有

$$\Delta_{h,k}u_{i,j}=-\left[\frac{1}{h^2}(u_{i-1,j}-2u_{i,j}+u_{i+1,j})+\frac{1}{k^2}(u_{i,j-1}-2u_{i,j}+u_{i,j+1})\right]$$
$$=0. \tag{7-5}$$

若取 $h=k$，差分格式可进一步简化为

$$\Delta_h u_{i,j}=-\frac{1}{h^2}(u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}-4u_{i,j})=0, \tag{7-6}$$

截断误差为 $O(h^2)$.

若选用(i,j),$(i+1,j+1)$,$(i-1,j+1)$,$(i+1,j-1)$和$(i-1,j-1)$五个节点，如图 7-3 所示，可构造另一五点差分格式(取 $h=k$)

$$\Delta_h u_{i,j}=-\frac{1}{2h^2}(u_{i+1,j+1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i-1,j-1}-4u_{i,j})$$
$$=0, \tag{7-7}$$

截断误差也是 $O(h^2)$.

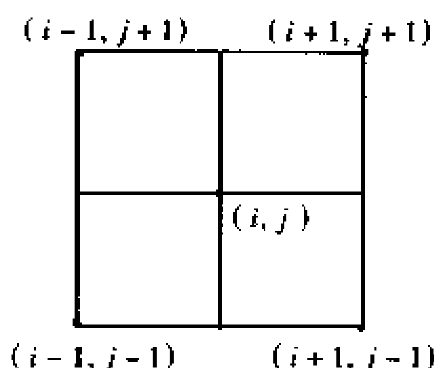


图 7-3

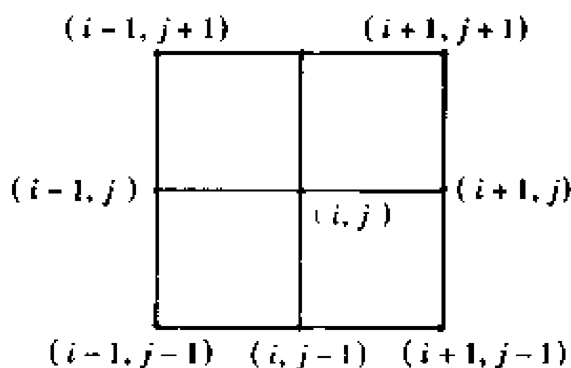


图 7-4

2. 九点差分格式

若用(7-6)和(7-7)式作线性组合，则可以得到精度较高的九点格式，用到的网格节点见图 7-4. 拉普拉斯**九点差分格式**为($h=k$)

$$\Delta_h u_{i,j}=-\frac{1}{6h^2}\{4(u_{i,j+1}+u_{i+1,j}+u_{i,j-1}+u_{i-1,j})+$$
$$(u_{i+1,j+1}+u_{i+1,j-1}+u_{i-1,j+1}+u_{i-1,j-1})-20u_{i,j}\}=0, \tag{7-8}$$

截断误差为 $O(h^4)$.

对泊松方程，九点差分格式为($h=k$)

$$\Delta_h u_{i,j}=-\frac{1}{6h^2}\{4(u_{i,j+1}+u_{i+1,j}+u_{i,j-1}+u_{i-1,j})+$$
$$(u_{i+1,j+1}+u_{i+1,j-1}+u_{i-1,j+1}+u_{i-1,j-1})-20u_{i,j}\}$$

$$= f_{i,j} + \frac{1}{12}(f_{i,j+1} + f_{i+1,j} + f_{i-1,j} + f_{i,j-1} - 4f_{i,j}), \quad (7\text{-}9)$$

截断误差为 $O(h^4)$.

7.2.3　边界处理

边界处理是利用边界条件对非正则内点，即 $G_h^{(2)}$ 上的点补充方程，使得到的差分方程组的数目和未知数的数目相等.

1. 第 I 边界条件

(1) 直接转移. 在非正则内点 P 上，不列差分方程，而是用边界上离这个点最近的边界点的值来代替. 如图 7-5，令

$$u_P = \bar{u}(R), \quad (7\text{-}10)$$

用(7-10)式近似，截断误差的阶为 $O(h)$.

(2) 线性插值. 如图 7-5，P 点的近似值可取

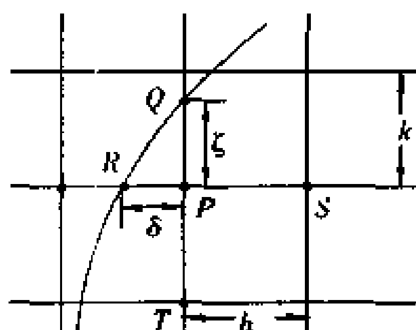


图 7-5

$$u_P = \frac{h}{h+\delta}u_R + \frac{\delta}{h+\delta}u_S, \quad (7\text{-}11)$$

或

$$u_P = \frac{k}{k+\xi}u_Q + \frac{\xi}{k+\xi}u_T, \quad (7\text{-}12)$$

截断误差为 $O(h^2)$.

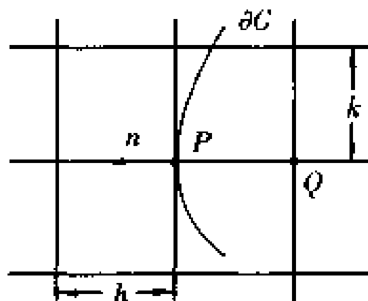


图 7-6

2. 对第 II、第 III 边界条件

当 P 为非正则内点时，对第 II、第 III 边界条件逼近的主要问题是处理$\frac{\partial u}{\partial n}$.

(1) P 点位于 ∂G 上，而且外法向 n 与坐标轴平行，如图 7-6，则

$$\left.\frac{\partial u}{\partial n}\right|_P = \frac{u_P - u_Q}{h} + O(h),$$

近似方程为

$$\frac{u_P - u_Q}{h} + \eta_P u_P = q_P, \quad (7\text{-}13)$$

截断误差为 $O(h)$.

(2) P 点位于 ∂G 上，但外法线方向 n 不与坐标轴平行，如图 7-7，则近似方程为

$$\left(\frac{u_P - u_Q}{h}\cos\alpha + \frac{u_P - u_R}{k}\sin\alpha\right) + \eta_P u_P = q_P, \quad (7\text{-}14)$$

截断误差为 $O(h)$.

(3) 非正则内点 P 不在 ∂G 上，如图 7-8，仍可按上面办法在 P 点列方程，但用边界 ∂G 上与 P 点相近的某点 P^* 的外法线方向作为 P 点的外法向 n，并把方程中

的 P 用 P^* 代替.

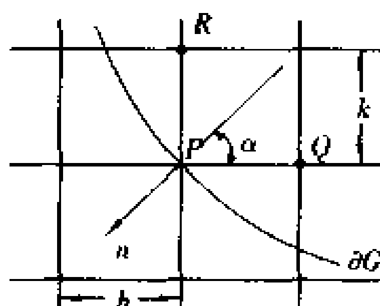


图 7-7

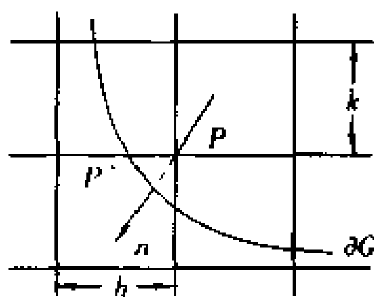


图 7-8

7.3　差分方程组的可解性和收敛性

在偏微分方程中,调和函数有极值原理,对应的差分方程也有类似性质. 下面以二阶椭圆型方程第 1 边值问题为例来讨论.

全部节点按一定次序编号,对第 i 个节点,把它周围所有相邻节点的集合记为 $U(i)$,这样前面介绍的差分格式可以统一表示为

$$\begin{cases} \Delta_h v_i = a_{ii}v_i - \sum\limits_{\substack{j\in U(i)\\ j\neq i}} a_{ij}v_j = f_i, & i\in G_h, \\ v_i = \alpha_i, & i\in \partial G_h. \end{cases} \tag{7-15}$$

并约定,求和号的下标 $j\in\partial G_h$ 时,将 $v_j = \alpha_j$ 的项移到方程右端,因此可令 $a_{ij} = 0$.

假定(7-15) 式的系数满足条件

$$\begin{cases} a_{ii} > 0, a_{ij} \geqslant 0, & i,j\in G_h, \\ d_{ii} = a_{ii} - \sum\limits_{\substack{j\in U(i)\\ j\neq i}} a_{ij} \geqslant 0, & i\in G_h, \end{cases} \tag{7-16}$$

并且存在 i_0,使当 $i = i_0$ 有一个相邻节点是边界点时,严格的不等号成立.

另外,区域 $\overline{G}_h = G_h\cup\partial G_h$ 是连通的,即对区域中任意两点 P_0、P_k,必有一串属于 G_h 的节点 $P_i(i = 1,2,\cdots,m)$ 可将 P_0、P_k 排列起来,如 $P_0P_1\cdots P_mP_k$,使前后两点为相邻节点.

差分格式(7-15) 的定义域 $\overline{G}_h$ 连通,系数 a_{ij}、a_{ii} 满足条件(7-16) 式,则有下面定理 1 ~ 定理 3 成立.

定理 1　(极值原理) 设

1° v_i 是定义在网格区域 $\overline{G}_h$ 上的一组值,

2° $v_i \neq$ 常数;

3° $\Delta_h v_i \leqslant 0, i\in G_h$;

则 v_i 不可能在内部节点上达到正的最大值.

类似地,如果将 3° 改为 $\Delta_h v_i \geqslant 0$,则 v_i 不可能在内部节点上达到负的最小值.

定理 2　差分格式(7-15) 的解存在而且唯一.

定理 3 （比较定理）设 w_i 和 v_i 是定义在网格 $\bar{G}_h$ 上的两组值，且

1° $\Delta_h v_i \geqslant |\Delta_h w_i|, i \in G_h$;

2° $v_i \geqslant |w_i|, i \in \partial G_h$,

则 $v_i \geqslant |w_i|, i \in \bar{G}_h$.

定理 4 若泊松方程第 Ⅰ 边界条件的解 $u(x,y) \in C^4(\bar{G})$，则五点差分格式的解 u_i 一致收敛到 $u(i)$，且有估计式

$$\max_{i \in G_h} |u(i) - u_i| \leqslant \max_{i \in \partial G_h} |\alpha_i| + Mh_1^2,$$

其中 M 是与 h 和 k 无关的常数，$h_1^2 = h^2 + k^2$，且 $\max\limits_{i \in \partial G_h} |\alpha_i|$ 是边界上逼近的最大误差.

定理 4 是先验估计，用起来有一定困难，下面介绍一种事后估计的办法.

设分别用 h、k 和 $h/2$、$k/2$ 算出了差分方程的解，并记为 $u_i(h,k)$ 和 $u_i(h/2, k/2)$，采用的差分格式具有截断误差 $O(h^r + k^r)$，则有

$$u_i(h,k) - u(i) \approx c(h^r + k^r),$$

$$u_i(h/2,k/2) - u(i) \approx c((h/2)^r + (k/2)^r),$$

其中 c 是与 h、k、r 无关的常数. 消去 c 得

$$u(i) - u_i\left(\frac{h}{2},\frac{k}{2}\right) \approx \frac{1}{2^r - 1}\left(u_i\left(\frac{h}{2},\frac{k}{2}\right) - u_i(h,k)\right).$$

一般情况下 r 总是大于 1 的，所以只要在计算过程中 $u_i(h,k)$ 和 $u_i(h/2,k/2)$ 的差小于容许误差，即可认为 $u_i(h/2,k/2)$ 是满足精度要求的近似解.

用差分法解椭圆型方程边界问题，离散后得到的是一线性代数方程组，求解过程可参阅数值代数部分.

8　抛物型方程的差分解法

8.1　抛物型方程及定解条件

8.1.1　方程

抛物型方程最常见的是热传导方程

$$Lu = \frac{\partial u}{\partial t} - b\frac{\partial^2 u}{\partial x^2} = 0 \quad (b > 0,\ 0 < t \leqslant T). \tag{8-1}$$

8.1.2　定解条件

1. 初值问题

若 x 的变化范围是 $-\infty < x < +\infty$，则方程(8-1)的定解条件只有初始条件：

$$u(x,0)=\varphi(x),\quad -\infty<x<+\infty. \tag{8-2}$$

若 $\varphi(x)$ 是 $|x|<+\infty$ 上给定的有界连续函数，本问题的解存在而且唯一，并连续依赖于初值.

2. 混合型问题

若 x 的变化范围是一个有限区域，不妨假设为 $0\leqslant x\leqslant 1$，则定解条件除给定初始条件外，还要在端点 $x=0,x=1$ 处增加边界条件.

第Ⅰ类边界条件：

$$\begin{cases}u(x,0)=\varphi(x), & 0\leqslant x\leqslant 1,\\ u(0,t)=g_1(t), & t\geqslant 0,\\ u(1,t)=g_2(t), & t\geqslant 0.\end{cases} \tag{8-3}$$

第Ⅱ、第Ⅲ类边界条件：

$$\begin{cases}u(x,0)=\varphi(x), & 0\leqslant x\leqslant 1,\\ \left(\dfrac{\partial u}{\partial x}-\lambda_1(t)u\right)_{x=0}=g_1(t), & t\geqslant 0,\\ \left(\dfrac{\partial u}{\partial x}+\lambda_2(t)u\right)_{x=1}=g_2(t), & t\geqslant 0,\end{cases} \tag{8-4}$$

其中 $\varphi(x)$、$g_1(t)$、$g_2(t)$、$\lambda_1(t)\geqslant 0$、$\lambda_2(t)\geqslant 0$ 都是给定的函数. 在(8-4)式中，当 $\lambda_1(t)$ 和 $\lambda_2(t)$ 都等于零时，为第Ⅱ类边界条件；当 $|\lambda_1(t)|+|\lambda_2(t)|\neq 0$ 时为第Ⅲ类边界条件.

下面主要介绍混合型问题的差分解法，并假设 $\varphi(x)$、$g_1(t)$、$g_2(t)$ 在讨论的区域上足够光滑，以保证 $u(x,t)$ 在所考虑的区域内存在、唯一，且具有所需要的有界偏导数.

8.2 抛物型方程的差分近似

8.2.1 网格剖分

首先将区域 $G=\{0\leqslant x\leqslant 1,0<t\leqslant T\}$ 进行剖分，取 $h=1/N,\tau>0$ 作网格线：

$$x=x_i=ih,\quad i=0,1,2,\cdots,N,$$

$$t=t_j=j\tau,\quad j=0,1,2,\cdots,\left\lceil\frac{T}{\tau}\right\rceil,$$

称格点 (x_i,t_j) 为节点，简记为 (i,j)，τ、h 分别是时间步长和空间步长.

8.2.2 差分格式

(1) 若用数值微分公式

$$\left(\frac{\partial u}{\partial t}\right)_{(i,j)}=\frac{1}{\tau}(u(i,j+1)-u(i,j))+O(\tau), \tag{8-5}$$

和

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(i,j)}=\frac{1}{h^2}(u(i-1,j)-2u(i,j)+u(i+1,j))+O(h^2),\qquad(8\text{-}6)$$

代入方程(8-1),略去误差项 $O(\tau+h^2)$,并用 $u_{i,j}$ 表示 $u(i,j)$ 的近似值,得差分格式

$$\Delta_{h,\tau}u_{i,j}=\frac{1}{\tau}(u_{i,j+1}-u_{i,j})-\frac{b}{h^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j})=0,\qquad(8\text{-}7)$$

其中 $\Delta_{h,\tau}$ 是差分算子.用(8-7)式逼近微分算子 $\mathrm{L}u_{i,j}$,其截断误差为 $O(\tau+h^2)$.

(2) 若一阶导数用向后差商

$$\left(\frac{\partial u}{\partial t}\right)_{(i,j)}=\frac{1}{\tau}(u(i,j)-u(i,j-1))+O(\tau)\qquad(8\text{-}8)$$

近似,将(8-8)和(8-6)式代入方程(8-1),得另一差分格式

$$\Delta_{h,\tau}u_{i,j+1}=\frac{1}{\tau}(u_{i,j+1}-u_{i,j})-\frac{b}{h^2}(u_{i+1,j}-2u_{ij}+u_{i-1,j})=0.\qquad(8\text{-}9)$$

(8-9)式逼近 $\mathrm{L}u_{i,j}$ 的截断误差为 $O(\tau+h^2)$.

(3) 若一阶导数用中心差商

$$\left(\frac{\partial u}{\partial t}\right)_{(i,j)}=\frac{1}{2\tau}(u(i,j+1)-u(i,j-1))+O(\tau^2)\qquad(8\text{-}10)$$

近似,把(8-10)和(8-6)式代入方程(8-1),得差分格式

$$\Delta_{h\tau}u_{i,j}=\frac{1}{2\tau}(u_{i,j+1}-u_{i,j-1})-\frac{b}{h^2}(u_{i+1,j}-2u_{ij}+u_{i-1,j})=0.\qquad(8\text{-}11)$$

(8-11)式逼近 $\mathrm{L}u_{i,j}$ 的截断误差为 $O(\tau^2+h^2)$.

(4) 计算 $\frac{\partial^2 u}{\partial x^2}$ 在第 $j+\frac{1}{2}$ 层的值,用 j 和 $j+1$ 层上的算术平均值来近似,即

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(i,j+\frac{1}{2})}=\frac{1}{2}\left(\left(\frac{\partial^2 u}{\partial x^2}\right)_{(i,j+1)}+\left(\frac{\partial^2 u}{\partial x^2}\right)_{(i,j)}\right)+O(\tau^2).\qquad(8\text{-}12)$$

将(8-12)和(8-5)式代入方程(8-1),得六点格式

$$\Delta_{h\tau}u_{i,j}=\frac{1}{\tau}(u_{i,j+1}-u_{ij})-\frac{b}{2h^2}(u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1})-\frac{b}{2h^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j})=0,\qquad(8\text{-}13)$$

用(8-13)式逼近 $\mathrm{L}u_{i,j}$ 的截断误差为 $O(\tau^2+h^2)$.

8.2.3 边界处理

对第Ⅰ类边界条件(8-3),可根据给定的函数直接得到

$$\begin{cases}u_{i,0}=\varphi(ih), & i=0,1,\cdots,N,\\ u_{0,j}=g_1(i\tau), & \\ u_{N,j}=g_2(j\tau), & j=1,2,\cdots,\lceil\frac{T}{\tau}\rceil.\end{cases}\qquad(8\text{-}14)$$

对第Ⅱ、第Ⅲ类边界条件(8-4),在点$(0,t)$用向前差商,在点$(1,t)$用向后差

商,则边界的差分方程为

$$\begin{cases} \frac{1}{h}(u_{1j}-u_{0j})-\lambda_{1j}u_{0j}=g_{1j}, \\ \frac{1}{h}(u_{Nj}-u_{N-1,j})+\lambda_{2j}u_{Nj}=g_{2j}, \end{cases} \tag{8-15}$$

其中 λ_{1j}、λ_{2j}、g_{1j}、g_{2j} 表示 $\lambda_1(j)$、$\lambda_2(j)$、$g_1(j)$、$g_2(j)$ 的值,这种差分近似的截断误差为 $O(h)$.

8.3 几种常用差分格式

对混合型问题第Ⅰ类边界条件给出差分格式,并令步长比 $s=\frac{\tau}{h^2}$.

1. 显式格式

由(8-7)和(8-14)式给出

$$\begin{cases} u_{i,j+1}=u_{i,j}+bs(u_{i+1,j}-2u_{i,j}+u_{i-1,j}), \\ hN=1, i=1,2,\cdots,N-1, j=0,1,\cdots,\left[\frac{T}{\tau}\right]-1, \\ u_{i,0}=\varphi(ih), \qquad i=0,1,\cdots,N, \\ u_{0,j}=g_1(j\tau), \\ u_{N,j}=g_2(j\tau), \qquad j=1,2,\cdots,\left[\frac{T}{\tau}\right]. \end{cases} \tag{8-16}$$

写成向量形式为

$$\begin{cases} \boldsymbol{u}_{j+1}=\boldsymbol{A}\boldsymbol{u}_j+\boldsymbol{f}_j, \quad j=0,1,\cdots,\left[\frac{T}{\tau}\right], \\ \boldsymbol{u}_0=\boldsymbol{\varphi}, \end{cases} \tag{8-17}$$

其中

$$\boldsymbol{u}_j=\begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{N-1,j} \end{bmatrix}, \quad \boldsymbol{f}_j=\begin{bmatrix} bsg_1(j\tau) \\ 0 \\ \vdots \\ 0 \\ bsg_2(j\tau) \end{bmatrix}, \quad \boldsymbol{\varphi}=\begin{bmatrix} \varphi(h) \\ \varphi(2h) \\ \vdots \\ \varphi((N-1)h) \end{bmatrix}, \tag{8-18}$$

$$\boldsymbol{A}=\begin{bmatrix} 1-2bs & bs & 0 & \cdots & 0 & 0 \\ bs & 1-2bs & bs & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & bs & 1-2bs \end{bmatrix}. \tag{8-19}$$

以上格式在初始条件 $u_{i,0}$ 和边界条件 $u_{0,j}$、$u_{N,j}$ 已知后,可以逐个计算,所以叫显式格式.

2. 隐式格式

由(8-9)和(8-14)式给出

$$\begin{cases} u_{i,j+1} - bs(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) = u_{i,j}, \\ i = 1,2,\cdots,N-1, \quad j = 0,1,\cdots,\lceil \frac{T}{\tau} \rceil - 1, \\ u_{i,0} = \varphi(ih), \quad i = 0,1,\cdots,N, \\ u_{0,j} = g_1(j\tau), \\ u_{N,j} = g_2(j\tau), \end{cases} \quad j = 1,2,\cdots,\lceil \frac{T}{\tau} \rceil. \tag{8-20}$$

写成向量形式为

$$\begin{cases} \boldsymbol{B}\boldsymbol{u}_{j+1} = \boldsymbol{u}_j + \boldsymbol{f}_{j+1}, \quad j = 0,1,2,\cdots,\lceil \frac{T}{\tau} \rceil - 1; \\ \boldsymbol{u}_0 = \boldsymbol{\varphi}. \end{cases} \tag{8-21}$$

其中

$$\boldsymbol{B} = \begin{bmatrix} 1+2bs & -bs & 0 & \cdots & 0 & 0 \\ -bs & 1+2bs & -bs & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -bs & 1+2bs \end{bmatrix}. \tag{8-22}$$

以上格式在已知 $\boldsymbol{u}_j$ 计算 $\boldsymbol{u}_{j+1}$ 时要解一个代数方程组，所以叫隐式格式.

3. 六点格式

由(8-13)和(8-14)式给出

$$\begin{cases} u_{i,j+1} - 2bs(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) = u_{i,j} + 2bs(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}), \\ i = 1,2,\cdots,N-1, \quad j = 0,1,2,\cdots,\lceil \frac{T}{\tau} \rceil - 1, \\ u_{i,0} = \varphi(ih), \quad i = 0,1,\cdots,N, \\ u_{0,j} = g_1(j\tau), \\ u_{N,j} = g_2(j\tau), \end{cases} \quad j = 0,1,2,\cdots,\lceil \frac{T}{\tau} \rceil. \tag{8-23}$$

写成向量形式为

$$\begin{cases} (\boldsymbol{I} + \boldsymbol{B})\boldsymbol{u}_{j+1} = (\boldsymbol{I} + \boldsymbol{A})\boldsymbol{u}_j + \boldsymbol{f}_{j+1} + \boldsymbol{f}_j, \quad j = 0,1,2,\cdots,\lceil \frac{T}{\tau} \rceil - 1, \\ \boldsymbol{u}_0 = \boldsymbol{\varphi}, \end{cases} \tag{8-24}$$

其中 $\boldsymbol{I}$ 是 $N-1$ 阶单位矩阵.六点格式是隐式格式.

8.4 差分格式的稳定性

前面几种差分格式可统一表示为

$$\begin{cases} \boldsymbol{u}_{j+1} = \boldsymbol{H}\boldsymbol{u}_j + \boldsymbol{f}_{j+1} \quad j = 0,1,\cdots,\lceil \frac{T}{\tau} \rceil - 1, \\ \boldsymbol{u}_0 = \boldsymbol{\varphi}, \end{cases} \tag{8-25}$$

对不同格式，对应的 $\boldsymbol{H}$ 不同.如显式格式 $\boldsymbol{H} = \boldsymbol{A}$，隐式格式 $\boldsymbol{H} = \boldsymbol{B}^{-1}$，六点格式 $\boldsymbol{H} = (\boldsymbol{I} + \boldsymbol{B})^{-1}(\boldsymbol{I} + \boldsymbol{A})$，$\boldsymbol{H}$ 称为迭代矩阵.

8.4.1 定义和定理

假设边界值精确,初始层引入误差 ε_0.用 u^* 表示初始层存在误差时由(8-25)式得到的解,即 u^* 满足方程

$$\begin{cases} u_{j+1}^* = Hu_j^* + f_{j+1}, & j = 0,1,\cdots,\lceil \frac{T}{\tau} \rceil - 1, \\ u_0 = \varphi + \varepsilon_0. \end{cases} \tag{8-26}$$

令 $\varepsilon_j = u_j^* - u_j$,则 ε_j 应满足方程

$$\begin{cases} \varepsilon_j = H\varepsilon_{j-1}, & j = 1,2,\cdots,\lceil \frac{T}{\tau} \rceil, \\ \varepsilon_0, & (\text{初始误差}). \end{cases} \tag{8-27}$$

由此推出

$$\| \varepsilon_j \| \leqslant \| H^j \| \| \varepsilon_0 \|, \quad j = 1,2,\cdots,\lceil \frac{T}{\tau} \rceil. \tag{8-28}$$

定义1 若 ε_j 在一定范数下满足不等式

$$\| \varepsilon_j \| \leqslant c \| \varepsilon_0 \| \quad (j \geqslant 1), \tag{8-29}$$

则称差分格式稳定,其中 c 是与 h、τ 无关的常数.

定理1 差分格式(8-25)稳定的充分必要条件是存在与 h、τ 无关的常数 c,使得对任何 $j(0 \leqslant j \leqslant \lceil \frac{T}{\tau} \rceil)$ 有

$$\| H^j \| \leqslant c. \tag{8-30}$$

若设 $\lambda_1,\lambda_2,\cdots,\lambda_{N-1}$ 是 H 的特征值,用 $\rho(H)$ 表示 H 的谱半径,则有

定理2 差分格式(8-25)稳定的必要条件是:

$$\rho(H) \leqslant 1 + O(\tau). \tag{8-31}$$

稳定性的这个必要条件十分重要,在许多情况下也是充分条件(如 H 是正规矩阵:$HH^* = H^*H$).

8.4.2 几种常见格式的稳定性

1. 显式格式

这时 $H = A$,A 的特征值

$$\lambda_k = 1 - 4bs\sin^2\left(\frac{k\pi}{2N}\right), \quad k = 1,2,\cdots,N-1.$$

要使(8-31)式成立,必须而且只须

$$bs \leqslant \frac{1}{2}.$$

注意到 A 的对称性,$bs \leqslant \frac{1}{2}$ 也是格式稳定的充分条件.

2. 隐式格式

这时 $H = B^{-1}$,特征值为

$$\mu_k^{-1} = \left(1 + 4bs\sin^2\left(\frac{k\pi}{2N}\right)\right)^{-1}, \quad k = 1,2,\cdots,N-1.$$

无论 s 如何选取，均有 $0 < \mu_k^{-1} < 1$，因此，隐式格式是无条件稳定的.

3. 六点格式

这时 $\boldsymbol{H} = (\boldsymbol{I} + \boldsymbol{B})^{-1}(\boldsymbol{I} + \boldsymbol{A})$，特征值为

$$\left(1 - 2bs\sin^2\left(\frac{k\pi}{2N}\right)\right)\left(1 + 2bs\sin^2\left(\frac{k\pi}{2N}\right)\right)^{-1}, \quad k = 1,2,\cdots,N-1.$$

不论 s 如何选取，特征值均小于 1，因此，六点格式恒稳定.

8.5 差分格式的收敛性

收敛就是当步长 h、$\tau \to 0$ 时，在一定范数意义下，对任一点 (i,j)，差分方程的解 $u_{i,j}$ 趋向于微分方程的解 $u(i,j)$.

1. 显式格式

令 $\rho_{i,j} = u(i,j) - u_{i,j}$，则 $\rho_{i,j}$ 满足的方程是

$$\begin{cases} \boldsymbol{\rho}_{j+1} = \boldsymbol{A}\boldsymbol{\rho}_j + \tau\boldsymbol{e}_j, & j = 0,1,\cdots,\left\lceil\frac{T}{\tau}\right\rceil - 1, \\ \boldsymbol{\rho}_0 = 0, \end{cases} \tag{8-32}$$

其中 $\boldsymbol{\rho}_j = (\rho_{1,j},\rho_{2,j},\cdots,\rho_{N-1,j})^{\mathrm{T}}$，$\boldsymbol{e}_j = (e_{1,j},e_{2,j},\cdots,e_{N-1,j})^{\mathrm{T}}$，矩阵 $\boldsymbol{A}$ 的定义由(8-19)式给出，$e_{ij} = O(\tau + h^2)$. 由方程(8-32)得

$$\boldsymbol{\rho}_j = \tau(\boldsymbol{A}^{j-1}\boldsymbol{e}_0 + \boldsymbol{A}^{j-2}\boldsymbol{e}_1 + \cdots + \boldsymbol{A}\boldsymbol{e}_{j-2} + \boldsymbol{e}_{j-1}). \tag{8-33}$$

若取 $\|\boldsymbol{\rho}_j\|_\infty = \max\limits_{1\leqslant i\leqslant N-1}|\rho_{i,j}|$；$\|\boldsymbol{e}_j\|_\infty = \max\limits_{1\leqslant i\leqslant N-1}|e_{i,j}| = O(\tau + h^2)$，则

$$\|\boldsymbol{\rho}_j\|_\infty \leqslant \tau(\|\boldsymbol{A}\|_\infty^{j-1}\|\boldsymbol{e}_0\|_\infty + \|\boldsymbol{A}\|_\infty^{j-2}\|\boldsymbol{e}_1\|_\infty + \cdots + \|\boldsymbol{e}_{j-1}\|_\infty).$$

当 $0 < bs < \frac{1}{2}$，$\|\boldsymbol{A}\|_\infty = 1$，步长比 $s = \frac{\tau}{h^2} =$ 常数时有

$$\|\boldsymbol{\rho}_j\|_\infty \leqslant \tau j \cdot O(h^2) \leqslant T \cdot O(h^2).$$

其中 $0 \leqslant j \leqslant T/\tau$，因而

$$|\rho_{i,j}| = O(h^2).$$

若第 I 边值问题的解在区域 $G = \{0 \leqslant x \leqslant 1, 0 \leqslant t \leqslant T\}$ 内有连续偏导数 $\frac{\partial^4 u}{\partial x^4}$、$\frac{\partial^2 u}{\partial t^2}$，并且 $0 < bs \leqslant \frac{1}{2}$，则显式差分格式收敛.

2. 隐式格式

隐式格式的误差满足方程

$$\boldsymbol{B}\boldsymbol{\rho}_{j+1} = \boldsymbol{\rho}_j + \tau\boldsymbol{e}_j, \quad \boldsymbol{\rho}_0 = \boldsymbol{0},$$

其中 $\boldsymbol{B}$ 由(8-22)式所定义. 隐式格式的截断误差 $\boldsymbol{e}_j = O(\tau + h^2)$. 因 $\boldsymbol{B}^{-1}$ 存在，故而有

$$\boldsymbol{\rho}_{j+1} = \tau(\boldsymbol{B}^{-1}\boldsymbol{e}_j + (\boldsymbol{B}^{-1})^2\boldsymbol{e}_{j-1} + \cdots + (\boldsymbol{B}^{-1})^{j+1}\boldsymbol{e}_0).$$

当 $h,\tau \to 0$ 时，$\|\boldsymbol{B}^{-1}\|_2 \leqslant 1$，$\|\boldsymbol{e}_j\|_2 \leqslant N^{\frac{1}{2}} \cdot O(\tau + h^2)$，于是

$$\| \rho_{j+1} \|_2 \leqslant \tau(j+1) N^{\frac{1}{2}} \cdot O(\tau + h^2),$$

其中 $Nh = 1, s = \tau/h^2 =$ 常数.即得

$$| \rho_{i,j} | = O(h^{3/2}).$$

若第1边值问题的解在区域 $G = \{0 \leqslant x \leqslant 1, 0 \leqslant t \leqslant T\}$ 内有连续偏导数 $\frac{\partial^4 u}{\partial x^4}$、$\frac{\partial^2 u}{\partial t^2}$,则不论 s 如何选取,隐式格式收敛.

3. 六点格式

六点格式的误差满足方程

$$(\boldsymbol{I} + \boldsymbol{B}) \boldsymbol{\delta}_{j+1} = (\boldsymbol{I} + \boldsymbol{A}) \boldsymbol{\rho}_j + \tau \boldsymbol{e}_j, \qquad \boldsymbol{\rho}_0 = \boldsymbol{0},$$

其中 $\boldsymbol{I}$ 是 $N-1$ 维单位矩阵,$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 由(8-19)、(8-22)式所定义,$\boldsymbol{e}_j = O(\tau^2 + h^2)$ 是六点格式的截断误差,因 $(\boldsymbol{I}+\boldsymbol{B})^{-1}$ 存在,$(\boldsymbol{I}+\boldsymbol{B})^{-1}(\boldsymbol{I}+\boldsymbol{A})$ 的特征值小于1,所以 $\| (I+B)^{-1}(I+A) \|_2 \leqslant 1$,于是 $\| \boldsymbol{\rho}_j \|_2 \leqslant T \cdot O(h^{-\frac{1}{2}}(\tau^2 + h^2))$,当步长比 $s = \frac{\tau}{h^2} =$ 常数时,得

$$| \rho_{i,j} | = O(h_h^{\frac{\tau}{2}}),$$

六点格式收敛.

一维抛物型方程 $\frac{\partial u}{\partial t} - b\frac{\partial^2 u}{\partial x^2} = 0$ 的差分格式如表8-1所示,其中 $b \geqslant 0$ 为常数,

$$\delta^2 u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j}, s = \tau/h^2.$$

表 8-1

名　称	差分格式	截断误差	稳定条件
(1) 显式	$u_{i,j+1} = u_{i,j} + bs\delta^2 u_{i,j}$	$O(\tau + h^2)$	$bs \leqslant \frac{1}{2}$
(2) 隐式	$u_{i,j+1} = u_{i,j} + bs\delta^2 u_{i,j+1}$	$O(\tau + h^2)$	恒稳
(3) 六点格式	$u_{i,j+1} = u_{i,j} + \frac{bs}{2}(\delta^2 u_{i,j} + \delta^2 u_{i,j+1})$	$O(\tau^2 + h^2)$	恒稳
(4) 菱形格式	$u_{i,j+1} = u_{i,j-1} + 2bs(u_{i+1,j} - u_{i,j+1} - u_{i,j-1} + u_{i-1,j})$	$O(\tau^2 + h^2) + O(\tau^2/h^2)$	恒稳
(5) 加权隐式 I	$\frac{3}{2}(u_{i,j+1} - u_{i,j}) - \frac{1}{2}(u_{i,j} - u_{i,j-1}) = bs\delta^2 u_{i,j+1}$	$O(\tau^2 + h^2)$	恒稳
(6) 加权隐式 II	$(1+\sigma)(u_{i,j+1} - u_{i,j}) - \sigma(u_{i,j} - u_{i,j-1}) = bs\delta^2 u_{i,j+1}$ $\sigma =$ 常数 $\geqslant 0$	$O(\tau + h^2)$	恒稳

续表

名 称	差分格式	截断误差	稳定条件
(7) 加权隐式 Ⅲ	格式同(6) 取 $\sigma = \frac{1}{2} - \frac{1}{12bs}$	$O(\tau^2 + h^4)$	恒稳
(8) 杜福尔-佛兰克尔格式	$\frac{1}{12}(u_{i+1,j+1} - u_{i+1,j}) + \frac{5}{6}(u_{i,j+1} - u_{i,j}) + \frac{1}{12}(u_{i-1,j+1} - u_{i-1,j}) = \frac{bs}{2}(\delta^2 u_{i,j+1} + \delta^2 u_{i,j})$	$O(\tau^2 + h^4)$	恒稳
(9) 理查森格式	$u_{i,j+1} = u_{i,j-1} + 2bs\delta^2 u_{i,j}$	$O(\tau^2 + h^2)$	恒不稳

注：格式(4),(5),(6),(7) 是三层格式，第二层的值须用另外方法给出. 稳定性定义为：在一定范数下满足不等式 $\| \varepsilon_j \| \leqslant c(\| \varepsilon_0 \| + \| \varepsilon_1 \|)(j \geqslant 2)$，其中 c 与 h 和 τ 无关.

8.6 二维热传导方程混合型问题的差分近似

以下列方程为例，介绍二维热传导方程的差分格式.

设

$$
\begin{cases}
\dfrac{\partial u}{\partial t} = b\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right), & 0 < x < 1, 0 < y < 1, \quad 0 < t < T, b > 0, \\
u(x,y,0) = f(x,y), & 0 \leqslant x \leqslant 1, 0 \leqslant y \leqslant 1, \\
u(0,y,t) = \psi_1(y,t), & \\
u(1,y,t) = \psi_2(y,t), & 0 \leqslant y \leqslant 1, 0 \leqslant t \leqslant T, \\
u(x,0,t) = \varphi_1(x,t), & \\
u(x,1,t) = \varphi_2(x,t), & 0 \leqslant x \leqslant 1, 0 \leqslant t \leqslant T,
\end{cases}
\tag{8-34}
$$

求在区域 $G = \{0 \leqslant x, y \leqslant 1, 0 \leqslant t \leqslant T\}$ 内满足方程和边界条件的函数 $u(x,y,t)$，其中 f、ψ_1、ψ_2、φ_1、φ_2 为已知连续函数.

取 $h_1 = \Delta x = 1/N, h_2 = \Delta y = 1/M, \Delta\tau = t > 0$，网格线为

$$
\begin{aligned}
x &= x_i = ih_1, \quad i = 0,1,2,\cdots,N, \\
y &= y_k = kh_2, \quad k = 0,1,2,\cdots,M, \\
t &= t_j = j\tau, \quad j = 0,1,2,\cdots,\left[\frac{T}{\tau}\right].
\end{aligned}
$$

用 (i,k,j) 表示空间的一个点 (x_i, y_k, t_j)，在点 (i,k,j) 上的函数值以 $u(i,k,j)$ 表示，近似值记为 $u_{i,k}^j$，并引进记号

$$
\begin{aligned}
\delta_x^2 u_{i,k}^j &= u_{i-1,k}^j - 2u_{i,k}^j + u_{i+1,k}^j, \\
\delta_y^2 u_{i,k}^j &= u_{i,k-1}^j - 2u_{i,k}^j + u_{i,k+1}^j, \\
s_1 &= \tau / h_1^2, \\
s_2 &= \tau / h_2^2.
\end{aligned}
$$

现将常用的二维热传导方程的差分格式列于表 8-2 中.

表 8-2

名　　称	差分格式	截断误差	稳定条件
(1) 显式格式	$u_{i,k}^{j+1} = u_{i,k}^{j} + b(s_1\delta_x^2 u_{i,k}^{j} + s_2\delta_y^2 u_{i,k}^{j})$	$O(\tau + h_1^2 + h_2^2)$	$\tau \leqslant \left[2b\left(\frac{1}{h_1^2} + \frac{1}{h_2^2}\right)\right]^{-1}$ 稳定
(2) 全隐式	$u_{i,k}^{j+1} = u_{i,k}^{j} + b(s_1\delta_x^2 u_{i,k}^{j+1} + s_2\delta_y^2 u_{i,k}^{j+1})$	$O(\tau + h_1^2 + h_2^2)$	恒稳
(3) 平均隐式	$u_{i,k}^{j+1} = u_{i,k}^{j} + \frac{b}{2}(s_1\delta_x^2 u_{i,k}^{j} + s_2\delta_y^2 u_{i,k}^{j}) + \frac{b}{2}(s_1\delta_x^2 u_{i,k}^{j+1} + s_2\delta_y^2 u_{i,k}^{j+1})$	$O(\tau + h_1^2 + h_2^2)$	恒稳
(4) 菱形格式	$\frac{1}{2}(u_{i,k}^{j+2} - u_{i,k}^{j})$ $= bs_1(u_{i-1,k}^{j+1} - u_{i,k}^{j+2} - u_{i,k}^{j} + u_{i+1,k}^{j+1}) + bs_2(u_{i,k-1}^{j+1} - u_{i,k}^{j+2} - u_{i,k}^{j} + u_{i,k+1}^{j+1})$	$O(\tau^2 + h_1^2 + h_2^2) + O\left(\frac{\tau^2}{h_1^2} + \frac{\tau^2}{h_2^2}\right)$	恒稳
(5)P-R 格式	$u_{i,k}^{j+1/2} = u_{i,k}^{j} + \frac{b}{2}(s_1\delta_x^2 u_{i,k}^{j+1/2} + s_2\delta_y^2 u_{i,k}^{j})$ $u_{i,k}^{j+1} = u_{i,k}^{j+1/2} + \frac{b}{2}(s_1\delta_x^2 u_{i,k}^{j+1/2} + s_2\delta_y^2 u_{i,k}^{j+1})$	$O(\tau^2 + h_1^2 + h_2^2)$	恒稳
(6)D-R 格式	$u_{i,k}^{j+\frac{1}{2}} = u_{i,k}^{j} + b(s_1\delta_x^2 u_{i,k}^{j+1/2} + s_2\delta_y^2 u_{i,k}^{j})$ $u_{i,k}^{j+1} = u_{i,k}^{j+1/2} + bs_2(\delta_y^2 u_{i,k}^{j+1} + \delta_y^2 u_{i,k}^{j})$	$O(\tau^2 + h_1^2 + h_2^2)$	恒稳
(7)C-N 格式	$u_{i,k}^{j+1} = u_{i,k}^{j} + \frac{bs_2}{2}(\delta_y^2 u_{i,k}^{j} + \delta_y^2 u_{i,k}^{j+1}) + \frac{bs_1}{2}(\delta_x^2 u_{i,k}^{j+1} + \delta_x^2 u_{i,k}^{j}) + \frac{b}{4}s_1 s_2 \delta_x^2 \delta_y^2 (u_{i,k}^{j+1} - u_{i,k}^{j})$	$O(\tau^2 + h_1^2 + h_2^2)$	恒稳
(8) 局部一维	$u_{i,k}^{j+1/2} = u_{i,k}^{j} + bs_1\delta_x^2 u_{i,k}^{j+\frac{1}{2}}$ $u_{i,k}^{j+1} = u_{i,k}^{j} + bs_2\delta_y^2 u_{i,k}^{j+1}$	$O(\tau + h_1^2 + h_2^2)$	恒稳

9　双曲型方程的差分解法

9.1　双曲型方程及其定解条件

9.1.1　一阶双曲型方程的定解问题

一阶双曲型方程定解问题为

$$\begin{cases}\dfrac{\partial u}{\partial t}+a\dfrac{\partial u}{\partial x}=0, & t>0,\\ u(x,0)=\varphi(x), & |x|<+\infty.\end{cases}\tag{9-1}$$

其解 $u(x,t)=\varphi(x-at)$, $|x|<\infty$, $t>0$,在平面 x-t 上沿直线

$$x-at=c\quad(c\text{ 是常数})\tag{9-2}$$

其值保持不变.直线(9-2)叫做**特征线**.

如果把初始条件看作 $t=0$ 时的"波"形,当 $a>0$ 时,波沿着 $+x$ 方向传播.(图 9-1(a)),当 $a<0$ 时,波沿 $-x$ 方向传播(图 9-1(b)),而波形不变.

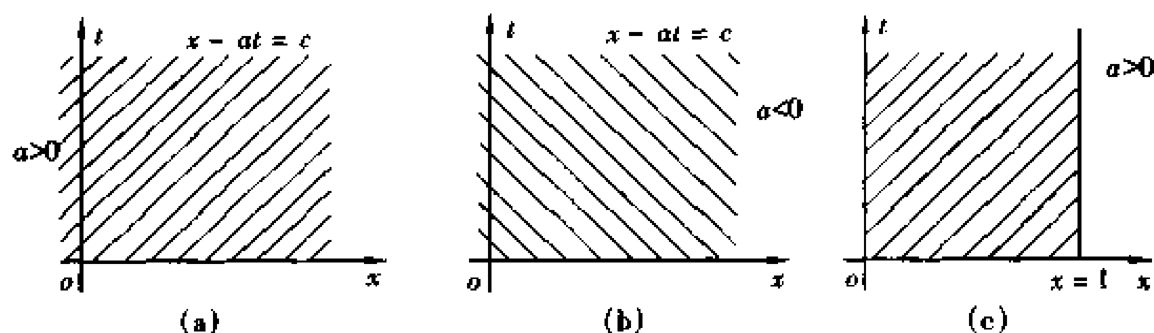


图 9-1

若初值问题局限在有界空间区域定解,则要规定恰当的边界条件.如图 9-1(c),$a>0$,在 $t=0$ 时给了初值,要求在 $t\geqslant 0, 0\leqslant x\leqslant 1$ 上的定解.设在左边界 $x=0$ 处给了 u 的边界条件,则由特征线走向可知,$t\geqslant 0, 0\leqslant x\leqslant 1$ 上的解唯一确定,在右边界 $x=1$ 处不能再给边界条件.反之,对 $a<0$ 的情形,则只能在右边界 $x=1$ 处给出边界条件.

9.1.2　二阶双曲型方程的定解问题

1. 方程

$$\frac{\partial^2 u}{\partial t^2}-a^2\frac{\partial^2 u}{\partial x^2}=0.\tag{9-3}$$

2.初始条件

$$\begin{cases} u\big|_{t=0} = \varphi(x), \\ \left.\dfrac{\partial u}{\partial t}\right|_{t=0} = \psi(x). \end{cases} \tag{9-4}$$

这时有两簇特征线：

$$x - at = c_1 \quad 和 \quad x + at = c_2 \quad (c_1, c_2 为常数).$$

3. 边界条件

方程(9-3)中含有 x 的二阶导数，定解问题应在左右两个边界上各给一个边界条件. 每一个边界的边界条件一般有三种给法：

第Ⅰ边界条件　给定函数值.

第Ⅱ边界条件　给定导数值.

第Ⅲ边界条件　给函数值和导数值的组合.

但不能只在一个边界上给出两个条件，这样的边界条件是不恰当的.

9.2 微分方程的差分近似

9.2.1 一阶双曲型方程的差分格式

将 x-t 平面用两组平行直线 $x = ih, t = j\tau$ 剖分成矩形网格，h 为空间步长，τ 为时间步长. 选用适当的差商来代替微商，就得到各种不同的差分格式，现将方程 $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$ 的各种差分格式列于表9-1，其中 $\beta = a\tau/h, \theta = nh, n$ 为任意实数.

表 9-1

名　称	差分格式	截断误差	增长因子和稳定条件
(1) 右偏显式	$u_{i,j+1} - u_{i,j} + \beta(u_{i+1,j} - u_{i,j}) = 0$	$O(\tau + h)$	$G = 1 + \beta(1 - e^{i\theta})$ $-1 \leqslant \beta \leqslant 0$ 稳定
(2) 左偏显式	$u_{i,j+1} - u_{i,j} + \beta(u_{i,j} - u_{i-1,j}) = 0$	$O(\tau + h)$	$G = 1 - \beta(1 - e^{-i\theta})$ $0 \leqslant \beta \leqslant 1$ 稳定
(3) 右偏隐式	$u_{i,j+1} - u_{i,j} + \beta(u_{i+1,j+1} - u_{i,j+1}) = 0$	$O(\tau + h)$	$G = (1 - \beta(1 - e^{i\theta}))^{-1}$ $\beta \leqslant 0$ 或 $\beta \geqslant 1$ 稳定
(4) 左偏隐式	$u_{i,j+1} - u_{i,j} + \beta(u_{i,j+1} - u_{i-1,j+1}) = 0$	$O(\tau + h)$	$G = (1 + \beta(1 - e^{-i\theta}))^{-1}$ $\beta \leqslant -1$ 或 $\beta \geqslant 0$ 稳定

续表

名　　称	差分格式	截断误差	增长因子和稳定条件
(5)中心差显式	$u_{i,j+1} - u_{i,j} + \frac{\beta}{2}(u_{i+1,j} - u_{i-1,j}) = 0$	$O(\tau + h^2)$	$G = 1 - \mathrm{i}\beta\sin\theta$ 恒不稳
(6)中心差隐式	$u_{i,j+1} - u_{i,j} + \frac{\beta}{2}(u_{i+1,j+1} - u_{i-1,j+1}) = 0$	$O(\tau + h^2)$	$G = (1 + \mathrm{i}\beta\sin\theta)^{-1}$ 恒稳
(7)平均隐式	$u_{i,j+1} - u_{i,j} + \frac{\beta}{4}(u_{i+1,j} - u_{i-1,j} + u_{i+1,j+1} - u_{i-1,j+1}) = 0$	$O(\tau + h^2)$	$G = \dfrac{1 - \frac{\mathrm{i}\beta}{2}\sin\theta}{1 + \frac{\mathrm{i}\beta}{2}\sin\theta}$ 恒稳
(8)拉克斯-温得罗夫(Lax-Wendroff)格式	$u_{i,j+1} - u_{i,j} + \frac{\beta}{2}(u_{i+1,j} - u_{i-1,j}) - \frac{\beta^2}{2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = 0$	$O(\tau^2 + h^2)$	$G = 1 - \frac{\beta}{2}(e^{\mathrm{i}\theta} - e^{-\mathrm{i}\theta}) + \beta^2(e^{\mathrm{i}\theta} - 2 + e^{-\mathrm{i}\theta})$ $\lvert\beta\rvert \leqslant 1$ 稳定
(9)菱形格式	$u_{i,j+2} - u_{i,j} + \beta(u_{i+1,j+1} - u_{i-1,j+1}) = 0$	$O(\tau^2 + h^2)$	$G_{1,2} = -\mathrm{i}\beta\sin\theta \pm (1 - \beta^2\sin^2\theta)^{1/2}$ $\lvert\beta\rvert \leqslant 1$,稳定
(10)拉克斯格式	$u_{i,j+1} - \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \frac{\beta}{2}(u_{i+1,j} - u_{i-1,j}) = 0$	$O(\tau + h^2) + O(h^2/\tau)$	$G = \cos\theta + \mathrm{i}\beta\sin\theta$ $\lvert\beta\rvert \leqslant 1$ 稳定

9.2.2　二阶双曲型方程

1. 矩形网格上二阶双曲型方程的差分格式

二阶双曲型方程：

$$\text{I}:\ \frac{\partial^2 u}{\partial t^2} - a^2\frac{\partial^2 u}{\partial x^2} = 0,$$

$$\text{II}:\ \frac{\partial^2 u}{\partial t^2} - a^2\frac{\partial^2 u}{\partial x^2} + b\frac{\partial u}{\partial x} + cu = 0.$$

其各种差分格式如表 9-2 所示，其中

$$\delta_t^2 u_{i,j+1} = u_{i,j+2} - 2u_{i,j+1} + u_{ij},\quad \delta_x^2 u_{ij} = u_{i-1,j} - 2u_{ij} + u_{i+1,j},$$

$$a > 0,\quad b, c \geqslant 0,\quad \beta = a\tau/h,\quad \theta = nh, \alpha = b\tau^2/h, \gamma = c\tau^2.$$

表 9-2

方程 Ⅰ

名　　称	差分格式	截断误差	特征方程和稳定条件
(1) 显式格式	$\delta_t^2 u_{i,j+1} = \beta^2 \delta_x^2 u_{i,j+1}$	$O(\tau^2+h^2)$	$G^2+(4\beta^2\sin^2\frac{\theta}{2}-2)\cdot G+1=0$ $\lvert\beta\rvert\leqslant 1$,稳定
(2) 隐式格式	$\delta_t^2 u_{i,j+1} = \beta^2 \delta_x^2 u_{i,j+2}$	$O(\tau^2+h^2)$	$(1+4\beta^2\sin^2\frac{\theta}{2})G^2-2G+1=0$ 恒稳
(3) 平均隐式	$\delta_t^2 u_{i,j+1} = \frac{\beta^2}{2}(\delta_x^2 u_{i,j}+\delta_x^2 u_{i,j+2})$	$O(\tau^2+h^2)$	$(1+2\beta^2\sin^2\frac{\theta}{2})G^2-2G+(1+2\beta^2\sin^2\frac{\theta}{2})=0$ 恒稳
(4) 九点格式	$\delta t^2 u_{i,j+1} = \beta^2(\sigma\delta_x^2 u_{i,j+2}+(1-2\sigma)\cdot\delta_x^2 u_{i,j+1}+\sigma\delta_x^2 u_{i,j})$ $0\leqslant\sigma\leqslant 1$	$O(\tau^2+h^2)$	$G^2-2G(1-\frac{2B}{1+4\sigma B})+1=0$ $B=\beta^2\sin^2\frac{\theta}{2}$, $\frac{1}{4}\leqslant\sigma\leqslant 1$,稳定 $0\leqslant\sigma<\frac{1}{4}$, $\beta\leqslant\frac{1}{\sqrt{1-4\sigma}}$

方程 Ⅱ

名　　称	差分格式	截断误差	特征方程和稳定条件
(1) 显式格式	$\delta_t^2 u_{i,j+1} = \beta^2\delta_x^2 u_{i,j+1}-\frac{\alpha}{2}(u_{i,j+2}-u_{i,j})-cu_{i,j+1}$	$O(\tau^2+h^2)$	$(1+\frac{\alpha}{2})G^2+(4\beta\sin^2\frac{\theta}{2}-2+\gamma)G+(1-\frac{\alpha}{2})=0$ $\beta^2+\gamma/4\leqslant 1$,稳定
(2) 隐式格式	$\delta_t^2 u_{i,j+1} = \beta^2\delta_x^2 u_{i,j+2}-\frac{\alpha}{2}(u_{i,j+2}-u_{i,j})-cu_{i,j+2}$	$O(\tau^2+h^2)$	$(1+4\beta^2\sin^2\frac{\theta}{2}+\frac{\alpha}{2}+\gamma)G^2-2G+(1-\frac{\alpha}{2})=0$　恒稳

续方程 Ⅱ

格点图	差分格式	截断误差	特征方程和稳定条件
(3) 平均隐式	$\delta_t^2 u_{i,j+1} = \frac{1}{2}(\beta^2\delta_x^2 u_{i,j} - \gamma u_{i,j}) + \frac{1}{2}(\beta^2\delta_x^2 u_{i,j+2} - \gamma u_{i,j+1}) - \frac{\alpha}{2}(u_{i,j+2} - u_{i,j})$	$O(\tau^2 + h^2)$	$(1 + 2\beta^2\sin^2\frac{\theta}{2} + \frac{\gamma}{2} + \frac{\alpha}{2})G^2 - 2G + (1 + 2\beta^2\sin^2\frac{\theta}{2} + \frac{\gamma}{2} - \frac{\alpha}{2}) = 0$ 恒稳
(4) 九点格式	$\delta_t^2 u_{i,j+1} = \beta^2(\sigma\delta_x^2 u_{i,j+2} + (1 - 2\sigma)\delta_x^2 u_{i,j+2} + \sigma\delta_x^2 u_{i,j}) - \frac{\alpha}{2}(u_{i,j+2} - u_{i,j}) - \gamma u_{i,j+1}$ $0 \leqslant \sigma \leqslant 1$	$O(\tau^2 + h^2)$	$(1 + \frac{\alpha}{2} + 4\sigma B)G^2 + (4(1 - 2\sigma)B + \gamma + 2)G + 4\sigma B - \frac{\alpha}{2} + 1 = 0$ $B = \beta^2\sin^2\frac{\theta}{2}$ $1/4 \leqslant \sigma \leqslant 1$,稳定 $0 \leqslant \sigma \leqslant \frac{1}{4}$, $\beta \leqslant 1/\sqrt{1 - 4\sigma}$

2. 初始条件处理

在方程 Ⅰ 和 Ⅱ 中有 t 的二阶导数,因此,有两个初始条件,

$$u\Big|_{t=0} = \varphi(x) \quad 和 \quad \frac{\partial u}{\partial t}\Big|_{t=0} = \psi(x).$$

对第一个初始条件,用直接转移 $u_{i,0} = \varphi_i$.

对第二个初始条件,可利用

(1) 向前差商 $u_{i,1} = u_{i,0} + \tau\psi_i$,截断误差 $O(\tau)$.

(2) 中心差商$\frac{1}{2\tau}(u_{i,1} - u_{i,-1}) = \psi_i$,截断误差 $O(\tau^2)$.

公式中出现了 $u_{i,-1}$,要利用微分方程的差分近似与初始条件联立来消去 $u_{i,-1}$.

例 1　波动方程定解问题

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2}, t > 0, \\ u\big|_{t=0} = \varphi(x), \\ \frac{\partial u}{\partial t}\big|_{t=0} = \psi(x). \end{cases}$$

利用显式格式得差分方程

$$\begin{cases} u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = \dfrac{\tau^2}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}), \\ u_{i,0} = \varphi_i, \\ u_{i,1} - u_{i,-1} = 2\tau\psi_i. \end{cases}$$

令 $j = 0$,则有

$$u_{i,1} - 2u_{i,0} + u_{i,-1} = \frac{\tau^2}{h^2}(u_{i+1,0} - 2u_{i,0} + u_{i-1,0}),$$

和初始条件联立,消去 $u_{i,-1}$,得

$$u_{i,1} = \frac{1}{2}\left\{2\tau\psi_i + \frac{\tau^2}{h^2}(\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}) - 2\varphi_i\right\}.$$

由上例看出,初始条件用中心差商近似,其精确度高些.但要求微分方程在初始线段 $t = 0$ 上也成立,多增加了一些要求.

3. 边界条件处理

关于边界条件的差分近似,和抛物型方程的办法完全一致,可参阅第8章.

9.3 定义、定理和稳定性

定义1 当步长 $h,\tau \to 0$ 时,若差分格式的极限形式是微分方程,就称该差分**格式与微分方程相容**.

前面各表中介绍的差分格式,当 $h,\tau \to 0$ 时,其截断误差趋向于零,均和相应微分方程相容.

定义2 差分格式的**依赖区域**是指当用某一格式计算某一层上的函数值时,所用到的初始层上网格节点的所在区间.

依赖区域与所用的差分格式有关,也和计算的时间层有关.

例如,用右偏格式计算 P 点上的函数值 u_P 时,如图9-2(a)要用到初始线段 BC 上的网格节点上的值,BC 线段就是右偏显式在点 P 的依赖区域.

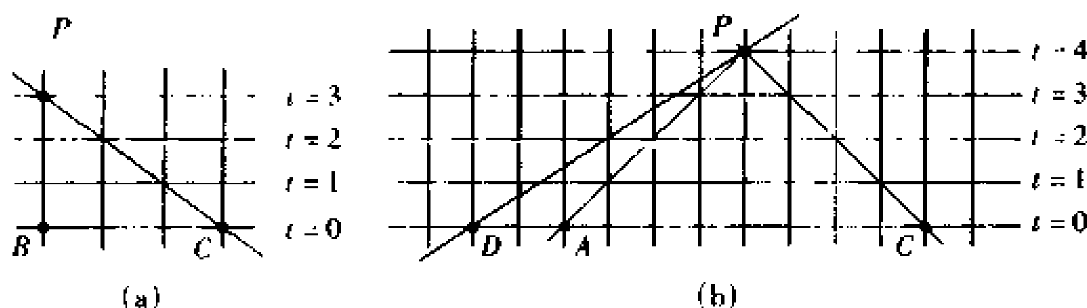


图 9-2

若用中心差显式计算 P 点上的函数值 u_P,如图9-2(b),要用到初始线段 AC 上网格节点,因此,中心差显式在 P 点的依赖区域为 AC 线段.

从 P 点引一条特征线,设与 $t = 0$ 交于 D 点,如图9-2(b).若 D 点在 AC 线段之外,说明差分格式的依赖区域不包含 D,因此,DA 线段上初始值的变化对用差分格

式计算 P 点的函数值 u_P 毫无影响，这时不能设想会有 $u_P - u(p) \to 0$.

定理1 （柯朗 - 弗里德里希斯 - 列维(Courant-Friedrichs-Lewy)条件）差分格式收敛的必要条件是差分格式的依赖区域包含微分方程的依赖区域. 柯朗 - 弗里德里希斯 - 列维条件简称**柯朗条件**或 **CFL 条件**.

例如当 $a \geqslant 0$ 时，左偏显式的柯朗条件是

$$0 \leqslant a\tau \leqslant h;$$

中心差显式的柯朗条件是

$$-h \leqslant Q\tau \leqslant h.$$

定理2 （拉克斯(Lax)等价定理）给定一个适定的线性微分方程初值问题和它的一个满足相容性的差分格式，则差分格式的稳定性是保证收敛的充要条件.

该定理说明，在一定条件下，差分格式稳定的结论等价于该格式的收敛性. 因此，下面只说明差分格式的稳定条件.

傅里叶(Fourier)方法是分析初值问题稳定性的一种常用方法，下面以右偏显式

$$u_{k,j+1} - u_{k,j} - \beta(u_{k+1,j} - u_{k,j}) = 0, \quad \beta = \tau a/h$$

为例，说明用傅里叶方法分析稳定性的过程.

设只在初始值 $u_{k,0}$ 上有误差 $\varepsilon_{k,0}$，即

$$(u + \varepsilon)_{k,0} = u_{k,0} + \varepsilon_{k,0},$$

则相应的解 $u_{k,j}$ 也有误差 $\varepsilon_{k,j}$，即

$$(u + \varepsilon)_{k,j} = u_{k,j} + \varepsilon_{k,j},$$

误差满足的方程为

$$\varepsilon_{k,j+1} - \varepsilon_{k,j} - \beta(\varepsilon_{k+1,j} - \varepsilon_{k,j}) = 0. \tag{9-5}$$

(1) 假设误差方程(9-5)的解是谐波形式

$$\varepsilon_{k,j} = G^j e^{inx_k} \quad (n \text{ 为任意实数}), \tag{9-6}$$

其中 G^j 为振幅，n 为频率.

(2) 将(9-6)式代入(9-5)式并消去公因子 $G^j e^{inx_k}$，并注意到 $x_k = kh, x_{k+1} = (k+1)h$，得到差分格式的特征方程

$$(G - 1) - \beta(e^{inh} - 1) = 0. \tag{9-7}$$

(3) 由(9-7)式算出增长因子 G

$$G = G(n) = (1 - 2\beta\sin^2\frac{nh}{2}) + i(2\beta\sin\frac{nh}{2}\cos\frac{nh}{2}). \tag{9-8}$$

(4) 由于初始误差可以表示为不同频率的谐波的叠加，并且由于计算中舍入误差的随机性，应该认为所有以 n 为频率的谐波分量都可能出现，因此，数值稳定的条件是

$$|G| \leqslant 1 + O(\tau)，\text{对一切 } n, 0 \leqslant j\tau \leqslant T. \tag{9-9}$$

(5) 从(9-8)式容易推出，若 $a > 0$, $-1 < \beta < 0$，则有 $|G| \leqslant 1$. 这就是右偏格式的稳定条件.

用同样方法可以导出表9-1和表9-2中各种格式的特征方程、增长因子和稳定

条件.

傅里叶方法原则上虽只适用于线性常系数方程,但可以适当推广到变系数以及非线性的情形.对变系数问题,要假定在一个小区间上不变,再在小区间上用傅里叶方法.对非线性问题则要逐段线性化.此外,因考虑初值问题,所以没有考虑边界条件的影响.若用傅里叶方法讨论混合型问题,则要注意边界条件,有时边界条件不影响稳定性的基本结论.

9.4　对流-扩散方程的差分格式及稳定条件

对流-扩散方程

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}-b\frac{\partial^2 u}{\partial x^2}=0,\qquad b\geqslant 0. \tag{9-10}$$

和前面一样建立差分格式,仍取 h 为空间步长,τ 为时间步长.选用适当的差商来代替微商,就得到各种差分格式如表9-3所示,其中

$$\delta_x^2 u_{i,j}=u_{i-1,j}-2u_{i,j}+u_{i+1,j},$$

$$\alpha=\frac{a\tau}{h},\quad \beta=\frac{b\tau}{h^2},\quad \theta=nh.$$

表 9-3

名　称	差分格式	截断误差	增长因子和稳定性条件
(1) 右偏显式	$u_{i,j+1}-u_{i,j}+\alpha(u_{i+1,j}-u_{i,j})-\beta\delta_x^2 u_{i,j}=0$	$O(\tau+h)$	$G=1-(2\beta-\alpha)\cdot(1-\cos\theta)-\mathrm{i}\alpha\sin\theta$ $0\leqslant 2\beta-\alpha\leqslant 1$, 并且 $\frac{\alpha^2}{\lvert 2\beta-\alpha\rvert}\leqslant 1$
(2) 左偏显式	$u_{i,j+1}-u_{i,j}+\alpha(u_{i,j}-u_{i-1,j})-\beta\delta_x^2 u_{i,j}=0$	$O(\tau+h)$	$G=1-(2\beta+\alpha)\cdot(1-\cos\theta)-\mathrm{i}\alpha\sin\theta$ $0\leqslant 2\beta+\alpha\leqslant 1$, 并且 $\frac{\alpha^2}{\lvert 2\beta+\alpha\rvert}\leqslant 1$
(3) 右偏隐式	$u_{i,j+1}-u_{i,j}+\alpha(u_{i+1,j+1}-u_{i,j+1})-\beta\delta_x^2 u_{i,j+1}=0$	$O(\tau+h)$	$G=(1+(2\beta-\alpha)\cdot(1-\cos\theta)+\mathrm{i}\alpha\sin\theta)^{-1}$ 当　$2\beta-\alpha\leqslant -1$ 或　$2\beta-\alpha\geqslant 0$
(4) 左偏隐式	$u_{i,j+1}-u_{i,j}+\alpha(u_{i,j+1}-u_{i-1,j+1})-\beta\delta_x^2 u_{i,j+1}=0$	$O(\tau+h)$	$G=(1+(2\beta+\alpha)\cdot(1-\cos\theta)+\mathrm{i}\alpha\sin\theta)^{-1}$ 当　$2\beta+\alpha\leqslant -1$ 或　$2\beta+\alpha\geqslant 0$

续表

名　　称	差分格式	截断误差	增长因子和稳定性条件
(5) 中心差显式	$u_{i,j+1}-u_{i,j}+\frac{\alpha}{2}(u_{i+1,j}-u_{i-1,j})-\beta\delta_x^2u_{i,j}=0$	$O(\tau+h^2)$	$G=1-2\beta(1-\cos\theta)-i\alpha\sin\theta$ $2\beta\leqslant 1$ 并且 $\frac{\alpha^2}{2\beta}\leqslant 1$
(6) 全隐式	$u_{i,j+1}-u_{i,j}+\frac{\alpha}{2}(u_{i+1,j+1}-u_{i-1,j+1})-\beta\delta_x^2u_{i,j+1}=0$	$O(\tau+h^2)$	$G=(1+2\beta(1-\cos\theta)+i\alpha\sin\theta)^{-1}$ 恒稳
(7) 平均隐式	$u_{i,j+1}-u_{i,j}-\frac{\beta}{2}(\delta_x^2u_{i,j+1}-\delta_x^2u_{i,j})+\frac{\alpha}{4}(u_{i+1,j+1}-u_{i-1,j+1}+u_{i+1,j}-u_{i-1,j})=0$	$O(\tau+h^2)$	$G=(1-\beta+\beta\cos\theta-i\frac{\alpha}{2}\sin\theta)\cdot(1+\beta-\beta\cos\theta+i\frac{\alpha}{2}\sin\theta)^{-1}$ 恒稳
(8) 拉克斯-温得罗夫格式	$u_{i,j+1}-u_{i,j}+\frac{\alpha}{2}(u_{i+1,j}-u_{i-1,j})-(\frac{\alpha^2}{2}+\beta)\delta_x^2u_{i,j}=0$	$O(\tau^2+h^2)$	$G=1-(\alpha^2+2\beta)+(\alpha^2+2\beta)\cos\theta-i\alpha\sin\theta$ $\alpha^2+2\beta\leqslant 1$
(9) 菱形格式	$u_{i,j+2}-u_{i,j}+\alpha(u_{i+1,j+1}-u_{i-1,j+1})-2\beta(u_{i-1,j+1}-u_{i,j+2}-u_{i,j}+u_{i+1,j+1})=0$	$O(\tau^2+h^2)+O(\frac{\tau^2}{h^2})$	$(1+2\beta)G^2-(2\beta\cos\theta-i\alpha\sin\theta)G-(1-2\beta)=0$ $\|\alpha\|<1$

参 考 文 献

1　徐萃薇. 计算方法引论. 北京:高等教育出版社,1985.

2　李庆扬,王能超,易大义. 数值分析. 武汉:华中工学院出版社,1981.

3　李荣华,冯果忱. 微分方程数值解法. 北京:人民教育出版社,1980.

4　陆金甫,关治. 偏微分方程数值解法. 北京:清华大学出版社,1987.

5　汤怀民,胡健伟. 微分方程数值方法. 天津:南开大学出版社,1990.

6　冯康. 数值计算方法. 北京:国防工业出版社,1978.

7　袁兆鼎,费景高,刘德贵. 刚性常微分方程初值问题的数值解法. 北京:科学出版社,1987.

·计算机数学卷·

第2篇

数值代数

编　者　白峰杉
审校者　刘　颖　蔡大用

目　　录

引　言

数值代数的基本问题是数值求解线性代数方程组

$$Ax = b,$$

和矩阵的特征值问题

$$Ax = \lambda x.$$

可以毫不夸张地讲,大部分的科学与工程计算问题最终要化成上述两类问题,特别是前者.其中的矩阵 A 可以是非奇异的也可以是奇异的,而且在很多情况下,它还是大规模和稀疏的.

这类问题看起来似乎简单,但在实际工作中,很多问题其本质的难点常常是如何针对问题的特点,选择适当的算法、运用合适的软件去有效地解决这类问题.

本篇通过古典算法的研究,力图体现求解这类问题算法的基本思想;同时从实用出发,介绍近十几年来发展成熟并得到广泛应用的若干新方法.

1　求解线性代数方程组的经典算法

数值求解线性代数方程组的方法,依其特点区分为直接法和迭代法两大类.所谓直接法是指,假设计算中没有舍入误差,经过有限次算术运算能给出问题精确解的数值方法.对中小规模的问题,这类方法很有效,而对大规模的问题,更多的要依靠迭代法.

直接法多是基于如下这个简单而又直观的事实:对于线性方程组

$$Rx = b, \tag{1-1}$$

如果系数矩阵 R 是上三角或者下三角的,则方程的解马上就可以得到.例如 R 是上三角的,方程(1-1) 即为

$$\begin{cases} r_{11}x_1 + r_{12}x_2 + \cdots + r_{1n}x_n = b_1, \\ \qquad\qquad r_{22}x_2 + \cdots + r_{2n}x_n = b_2, \\ \qquad\qquad\qquad \cdots\cdots \\ \qquad\qquad\qquad\qquad r_{nn}x_n = b_n, \end{cases}$$

只要矩阵 R 对角线上的元素 $r_{kk} \neq 0, k = 1,2,\cdots,n$,就可以依次求出 $x_n, x_{n-1},\cdots, x_2, x_1$.

经典的直接法,其基本思想就是设法将线性代数方程组化成与之等价的上三角(或下三角) 形式求解.

1.1　高斯消去法与 LU 分解

考虑一般的线性代数方程组

$$Ax = b, \tag{1-2}$$

其中矩阵 A 是可逆的.(1-2)的分量形式为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n. \end{cases}$$

1.1.1 高斯消去法

由于 A 可逆,故 $a_{k1}, k = 1,2,\cdots,n$,至少有一个不为零.设 $a_{11} \neq 0$,则第一次消元为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ \qquad a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \qquad \cdots\cdots \\ \qquad a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)}. \end{cases}$$

其中 $$a_{ij}^{(1)} = a_{ij} - l_{i1}a_{1j}, \quad b_j^{(1)} = b_j - l_{i1}b_1, l_{i1} = a_{i1}/a_{11},$$
$$i = 2,3,\cdots,n, \quad j = 2,\cdots,n.$$

还是由于 A 可逆,故 $a_{k2}^{(1)}, k = 2,\cdots,n$,中至少有一个非零.设 $a_{22}^{(1)} \neq 0$,可以进行类似的消元.一般地,设 $k-1$ 次消元后的结果为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k + \cdots + a_{1n}x_n = b_1, \\ \qquad a_{22}^{(1)}x_2 + \cdots + a_{2k}^{(1)}x_k + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \qquad\qquad \cdots\cdots \\ \qquad\qquad a_{kk}^{(k-1)}x_k + \cdots + a_{kn}^{(k-1)}x_n = b_k^{(n-1)}, \\ \qquad\qquad \cdots\cdots \\ \qquad\qquad a_{nk}^{(k-1)}x_k + \cdots + a_{nn}^{(k-1)}x_n = b_n^{(k-1)}. \end{cases} \tag{1-3}$$

设 $a_{kk}^{(k-1)} \neq 0$,第 k 次消元可以将后面 $n-k$ 个方程中的 x_k 项消去.所以经过 $n-1$ 次的消元得

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ \qquad a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \qquad\qquad \cdots\cdots \\ \qquad\qquad a_{nn}^{(n-1)}x_n = b_n^{(n-1)}. \end{cases} \tag{1-4}$$

这就得到一个与原方程组(1-2)等价的上三角方程组,其中 $a_{ij}^{(0)} = a_{ij}, b_j^{(0)} = b_j, i, j = 1,2,\cdots,n$,

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - l_{ik}a_{kj}^{(k-1)}, \quad b_i^{(k)} = b_i^{(k-1)} - l_{ik}b_k^{(k-1)}, l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}},$$
$$k = 1,2,\cdots,n-1, \qquad i,j = k+1,\cdots,n. \tag{1-5}$$

1.1.2 矩阵的 *LU* 分解

记消元后线性方程组(1-4)的系数矩阵为 U,它是一个上三角的. 再记单位下三角矩阵

$$L=\begin{bmatrix}1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1\end{bmatrix},$$

则方程组(1-2)的系数矩阵 A 有

$$A=LU.$$

称这为矩阵 A 的 LU 分解. 这时求解方程组(1-2)等价于顺序求解如下两个三角型的方程组:

$$Ly=b,\qquad Ux=y.$$

这与高斯消去法是等价的.

1.1.3 例子

必须注意的是,计算机的运算是有限字长的,这对算法会有很大影响. 如考虑方程组

$$\begin{cases}10^{-4}x_1+x_2=1,\\ \qquad x_1+x_2=2,\end{cases}$$

用高斯消去法消元一次得到

$$\begin{cases}10^{-4}x_1+x_2 \quad =1,\\ (1-10^4)x_2 \quad =2-10^4.\end{cases}$$

由此可得方程组的解

$$x_1=\frac{1}{1-10^{-4}},\quad x_2=1-\frac{10^{-4}}{1-10^{-4}}.$$

假设在一台计算机上求解上述方程,该计算机浮点数集为十进制且尾数长度为3,则上述方程在计算机中的形式为

$$0.100\text{E}-3\ x_1+0.100\text{E}1\ x_2=0.100\text{E}1,$$
$$0.100\text{E}1\quad x_1+0.100\text{E}1\ x_2=0.200\text{E}1.$$

仍然用第一个方程消去第二个方程中的 x_1 项,这时第二个方程变成

$$(0.100\text{E}5-0.100\text{E}1)x_2=(0.100\text{E}5-0.200\text{E}1),$$

由浮点数的运算规则,上述方程为

$$0.100\text{E}5\ x_2=0.100\text{E}5,$$

由此得到
$$x_2=1,\quad x_1=0.$$

显然这根本不是原方程的解.

如果用第二个方程消去第一个方程中含 x_1 的项,消去的结果为

$$\begin{cases} 0.100\text{E}1\ x_1 + 0.100\text{E}1\ x_2 = 0.200\text{E}1, \\ \qquad\qquad\quad 0.100\text{E}1\ x_2 = 0.100\text{E}1, \end{cases}$$

由此得到的计算解为 $x_2 = 1, x_1 = 1$. 可见得到了原方程解的很好的近似.

1.1.4 主元素高斯消去法

前面的高斯消去法第 k 步要求 $a_{kk}^{(k-1)} \neq 0$,该元素称为第 k 步的**主元素**. 在消去过程中即使 $a_{kk}^{(k-1)} \neq 0$,若绝对值比较小,也会因用它作除数引起舍入误差的急剧增长.

主元素消去法是对高斯消去法的改进. 如果第 k 步的主元选为

$$\left| a_{i_k k}^{(k-1)} \right| = \max_{k \leq j \leq n} \left| a_{jk}^{(k-1)} \right|,$$

对换第 k 个方程和 j_k 个方程后再执行消去,该方法称为**列主元高斯消去法**. 如果记初等矩阵

$$P(i,j) = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & 0 & \cdots & 1 & & & \\ & & \vdots & 1 \ddots 1 & \vdots & & & \\ & & 1 & \cdots & 0 & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ i \\ \\ j \\ \\ \\ \\ \end{matrix} \tag{1-6}$$

$$\qquad\qquad i \qquad\qquad j$$

则列主元高斯消去法对应的矩阵分解为

$$PA = LU.$$

而如果选主元为

$$\left| a_{i_k j_k}^{(k-1)} \right| = \max_{k \leq i,j \leq n} \left| a_{i,j}^{(k-1)} \right|,$$

则称消去法为**全主元高斯消去法**. 它所对应的矩阵分解则为

$$PAQ = LU,$$

其中矩阵 P 和 Q 是一系列形如矩阵(1-6)的乘积. 注意到,全主元消去法不仅要对换方程次序,还要交换未知数的位置.

1.1.5 对称正定矩阵的楚列斯基分解

如果 n 阶矩阵 A 是对称正定的,则 A 有如下的分解:

$$A = LDL^{\mathrm{T}}, \tag{1-7}$$

其中 D 是对角矩阵,L 是单位下三角矩阵. 由于 A 是正定的,所以 D 的所有对角元素都是正的. 记 D 的对角元素为 $d_1, d_2, \cdots, d_n$,则

$$D = D^{\frac{1}{2}} D^{\frac{1}{2}},$$

其中

$$D^{\frac{1}{2}}=\begin{bmatrix}\sqrt{d_1} & & & \\ & \sqrt{d_2} & & \\ & & \ddots & \\ & & & \sqrt{d_n}\end{bmatrix}.$$

由此及(1-7)式得到

$$A=(LD^{\frac{1}{2}})(LD^{\frac{1}{2}})^{\mathrm{T}}.$$

可见,当 A 对称正定时,则存在唯一的对角元素均为正数的下三角矩阵 L,使得

$$A=LL^{\mathrm{T}}. \tag{1-8}$$

(1-8)式称为 A 的**楚列斯基(Cholesky)分解**.所以求解以 A 为系数矩阵的线性代数方程组,可以通过顺序求解

$$Ly=b,\qquad L^{\mathrm{T}}x=y$$

得到.

1.1.6 算法的计算复杂度

若 A 是 n 阶非奇异矩阵,则高斯消去法求解方程组(1-2)需要计算

乘除次数:$n^3/3+n^2-n/3\approx\frac{1}{3}n^3$,

加减次数:$n(n-1)(2n+5)/6\approx\frac{1}{3}n^3$.

列主元高斯消去法还需要额外计算

逻辑运算次数:$n(n-1)/2$;

全主元高斯消去法需要计算

逻辑运算次数:$n(n-1)(2n-1)/6$.

1.2 矩阵条件数与消去法增长因子

1.2.1 向量范数

空间 $\mathbf{R}^n$ 中向量 x 的范数 $\|x\|$,是一个实值函数,它满足如下的条件:

1° 对所有的 $x\in\mathbf{R}^n$,都有 $\|x\|\geqslant 0$.而且当且仅当 $x=0$ 时,$\|x\|=0$;

2° 对任意的 $\alpha\in\mathbf{R}$ 和 $x\in\mathbf{R}^n$,都有

$$\|\alpha x\|=|\alpha|\,\|x\|;$$

3° 对所有的 x、$y\in\mathbf{R}^n$,都有

$$\|x+y\|\leqslant\|x\|+\|y\|.$$

记 $x=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\mathbf{R}^n$,常用的向量范数有

2-范数:$\|x\|_2=\left(\sum_{i=1}^{n}|x_i|^2\right)^{1/2}$;

1- 范数：$\|x\|_1 = \sum_{i=1}^{n} |x_i|$；

∞- 范数：$\|x\|_\infty = \max_{1\leqslant i\leqslant n} |x_i|$.

更一般的有 p- 范数

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

1.2.2 矩阵范数

矩阵 $A \in \mathbf{R}^{m\times n}$ 的范数 $\|A\|$，是一个实值函数，它满足如下的条件：

1° 对所有的 $A \in \mathbf{R}^{m\times n}$，都有 $\|A\| \geqslant 0$. 当且仅当 A 为零矩阵时，有 $\|A\| = 0$；

2° 对任意的 $\alpha \in \mathbf{R}$ 及任意的 $A \in \mathbf{R}^{m\times n}$，都有

$$\|\alpha A\| = |\alpha| \|A\|;$$

3° 对任意的 A、$B \in \mathbf{R}^{m\times n}$，有

$$\|A + B\| \leqslant \|A\| + \|B\|;$$

4° 对任意的 $A \in \mathbf{R}^{m\times n}$，$B \in \mathbf{R}^{n\times k}$，都有

$$\|AB\| \leqslant \|A\| \|B\|.$$

复向量或复矩阵范数的定义是完全类似的.

对 n 阶方阵 A，假设其特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_n$，则

$$\rho(A) = \max_{1\leqslant i\leqslant n} |\lambda_i|$$

称为矩阵 A 的谱半径. 对任意的一种范数，都有

$$\rho(A) \leqslant \|A\|.$$

另一方面对任意的 $\epsilon > 0$，都存在一种范数 $\|\cdot\|_\epsilon$（该范数与 ϵ 有关），使得

$$\|A\|_\epsilon \leqslant \rho(A) + \epsilon.$$

设 $A = [a_{ij}]_{m\times n}$，常用的矩阵范数有

2- 范数：$\|A\|_2 = (\rho(A^{\mathrm{T}}A))^{1/2}$；

1- 范数：$\|A\|_1 = \max_{1\leqslant j\leqslant n} \sum_{i=1}^{m} |a_{ij}|$；

∞- 范数：$\|A\|_\infty = \max_{1\leqslant i\leqslant m} \sum_{j=1}^{n} |a_{ij}|$.

它们分别与相应向量范数相容，即

$$\|Ax\|_s \leqslant \|A\|_s \|x\|_s, \quad s = 1, 2, \infty.$$

任意给定一种向量范数 $\|\cdot\|$，如下的函数

$$\|A\| = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|}$$

给出了一种矩阵范数，称之为**算子范数**，也称为该向量范数的**诱导范数**或**导出范数**. 矩阵的 2- 范数、1- 范数和 ∞- 范数，就分别是由相应的向量范数诱导出来的.

值得注意的是，不是算子范数的矩阵范数也是存在的，例如，下面著名的

Frobenius 范数就不能由任何一种向量范数导出.

$$\| A \|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} | a_{ij} |^2 \right)^{1/2}.$$

1.2.3　矩阵的条件数

假设 A 为 n 阶非奇异矩阵,则称

$$\mathrm{cond}(A) = \| A \| \| A^{-1} \|$$

为矩阵 A 的**条件数**.若 $\mathrm{cond}(A) \gg 1$,称 A 是坏条件的或病态的.

考虑矩阵

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix},$$

它的逆矩阵为

$$A^{-1} = \begin{bmatrix} 10001 & -10000 \\ -10000 & 10000 \end{bmatrix},$$

于是有

$$\mathrm{cond}_\infty(A) = \| A \|_\infty \| A^{-1} \|_\infty = 20001 \times 2.0001 = 40004.001,$$

所以 A 是病态的.希尔伯特矩阵

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

是典型的坏条件矩阵.

1.2.4　解的误差估计

误差是实际问题中不可避免的.通常模型有误差,观测有误差,计算也会有误差.误差对方程的解究竟有怎样的影响?先看方程

$$\begin{bmatrix} 2 & 6 \\ 2 & 6.00001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 8.00001 \end{bmatrix},$$

它的解是 $x = (1,1)^T$.将上述方程扰动为

$$\begin{bmatrix} 2 & 6 \\ 2 & 5.99999 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 8.00002 \end{bmatrix},$$

这时方程的解为 $x = (10, -2)^T$.这表明估计解的误差是十分重要的.

假设线性方程组(1-2)的系数矩阵和右端分别有扰动 δA 和 δb,由此引起解的扰动 δx,则

$$(A + \delta A)(x + \delta x) = b + \delta b.$$

如果 A 是可逆矩阵,向量 $b \neq 0$ 且 $\| A^{-1} \| \| \delta A \| < 1$,则

$$\frac{\|\delta x\|}{\|x\|} \leqslant \frac{\operatorname{cond}(A)}{1-\|A^{-1}\|\|\delta A\|}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right).$$

对于确定的相对误差 $\|\delta A\|/\|A\|$ 和 $\|\delta b\|/\|b\|$ 而言，cond(A)越小，就可以确保解的误差 $\|\delta x\|/\|x\|$ 比较小；如果 cond(A)很大，解的误差就可能很大.所以 cond(A) 刻划了方程解对原始数据变化的敏感程度.

实际计算中如下的估计更有意义.设 $\tilde{x}$ 是方程组(1-2) 的计算解，则

$$r = b - A\tilde{x}$$

称为**残差**.若 A 可逆，$b \neq 0$，则有

$$\frac{1}{\operatorname{cond}(A)}\frac{\|r\|}{\|b\|} \leqslant \frac{\|\tilde{x}-x\|}{\|x\|} \leqslant \operatorname{cond}(A)\frac{\|r\|}{\|b\|}.$$

上述结果给出了方程组计算解的误差界.在 A 的条件数已知(或有较好估计) 时，该误差界是可以计算的，这可以帮助我们分析计算解的可靠程度.

1.2.5　消去法的增长因子

假设方程组(1-2) 的计算解是 $\tilde{x}$，它可以看成是如下方程的精确解：

$$(A+E)\tilde{x} = b.$$

基于这一点可以给出消去法的**向后误差估计**：

$$\frac{\|\tilde{x}-x\|_\infty}{\|x\|_\infty} \leqslant 4n^2\operatorname{cond}_\infty(A)\rho\varepsilon, \tag{1-9}$$

其中 x 为方程组(1-2) 的精确解，ε 为计算机浮点运算精度，ρ 称为消去法的**增长因子**，它定义为

$$\rho = \frac{\max\limits_{i,j,k}|a_{ij}^{(k)}|}{\max\limits_{i,j}|a_{ij}|}.$$

这里的 a_{ij} 是矩阵 A 的元素，$a_{ij}^{(k)}$ 是 A 经过 k 次消元后的元素.

消去法的向后误差估计(1-9) 式表明，消去法的误差与增长因子的大小有密切关系.在其它因素不变的前提下，增长因子越大，数值方法的误差可能越大.

对列主元高斯消去法，

$$\rho \leqslant 2^{n-1},$$

而且这个上界是可以达到的.例如，对

$$A = \begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & 1 \end{bmatrix}.$$

对全主元高斯消去法，则有

$$\rho \leqslant \left(n \cdot 2^1 \cdot 3^{1/2} \cdot 4^{1/3} \cdots n^{1/(n-1)} \right)^{1/2}.$$

它在量级上要比列主元方法小.

上述增长因子的上界都是指数型的,特别是列主元高斯消去法,确有个别问题的增长因子能达到这个界.但一般来讲,增长因子的上界估计是十分保守的.在实际应用中,列主元的高斯消去法是常用的,而且通常是可靠的.这个问题困扰了人们很长时间,近来得到了在统计平均意义下高斯消去法增长因子的分析.全主元消去法的统计平均增长因子约为 $n^{1/2}$,而列主元高斯消去法仅有 $n^{2/3}$.这从一个侧面说明高斯消去法在统计平均意义下是可靠的.

1.2.6 矩阵条件数的估计

前面几节的分析表明,了解一个矩阵的条件数可以帮助我们研究计算解的可信程度.通过直接计算 A^{-1} 从而计算它的条件数显然不可行,只要能较好地估计它就可以了.

考虑方程 $Ax = b$.假设 A 是可逆的,则

$$x = A^{-1}b.$$

两边取范数,得到

$$\| x \| = \| A^{-1}b \| \leqslant \| A^{-1} \| \, \| b \|.$$

所以只要 $b \neq 0$,即有

$$\| A^{-1} \| \geqslant \frac{\| x \|}{\| b \|}.$$

取 k 个向量 $b^{(1)}, b^{(2)}, \cdots, b^{(k)}$,相应的解设为 $x^{(1)}, x^{(2)}, \cdots, x^{(k)}$,则有

$$\| A^{-1} \| \geqslant \max_{1 \leqslant p \leqslant k} \frac{\| x^{(p)} \|}{\| b^{(p)} \|}.$$

将 A 的条件数的估计值取为

$$\operatorname{cond}(A) \approx \| A \| \cdot \max_{1 \leqslant p \leqslant k} \frac{\| x^{(p)} \|}{\| b^{(p)} \|}.$$

这是许多现行软件中提供矩阵条件数估计的方法,通过对 b 的适当选取,可以给出矩阵条件数的较好估计.

1.3 QR 算法

在线性方程组的直接方法中,QR 算法比之高斯消去法具有可靠性更高的优点,同时 QR 方法所需计算量也高于高斯消去法.所谓 QR 方法基于矩阵的 QR 分解.这种矩阵分解很重要,在特征值的计算中还会用到.

1.3.1 矩阵的 QR 分解

一个方阵可以被分解为一个下三角矩阵和一个上三角矩阵的乘积,这是矩阵的 LU 分解.矩阵的 QR 分解是指它可以分解为一个正交矩阵和一个上三角矩阵的乘积,即

$$A = QR,\tag{1-10}$$

其中 Q 是正交矩阵，R 是上三角矩阵.

给出矩阵 QR 分解的途径有多种，这里介绍常用的豪斯霍尔德(Householder)方法. 对任意单位长度的 n 维向量 w(即 $w^{T}w = 1$)、矩阵

$$P = I_n - 2ww^{T}$$

一定是正交的. 设 $A = [a_{ij}]_{n\times n}$，取

$$w_1 = \mu_1(a_{11} - s_1, a_{21}, a_{31}, \cdots, a_{n1})^{T},$$

其中

$$s_1 = \left\{\sum_{j=1}^{n}(a_{j1})^2\right\}^{1/2},\quad \mu_1 = \frac{1}{\sqrt{2s_1(s_1 - a_{11})}}.$$

可以验证

$$A^{(1)} = (I_n - 2w_1w_1^{T})A = \begin{bmatrix} s_1 & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}.$$

再取

$$w_2 = \mu_2(0, a_{22}^{(1)} - s_2, a_{32}^{(1)}, \cdots, a_{n2}^{(1)})^{T}.$$

其中

$$s_2 = \left\{\sum_{j=2}^{n}(a_{j2}^{(1)})^2\right\}^{1/2},\quad \mu_2 = \frac{1}{\sqrt{2s_2(s_2 - a_{22}^{(1)})}}.$$

记 $P^{(2)} = I_n - 2w_2w_2^{T}$，则

$$A^{(2)} = P^{(2)}A^{(1)} = P^{(2)}P^{(1)}A = \begin{bmatrix} s_1 & \times & \cdots & \times \\ 0 & s_2 & \cdots & \times \\ 0 & 0 & \cdots & \times \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \times \end{bmatrix}.$$

继续这一过程，最终就得到

$$R = P^{(n-1)}P^{(n-2)}\cdots P^{(2)}P^{(1)}A$$

是上三角矩阵. 由于每个 $P^{(i)}$ 都是正交的，所以

$$Q = P^{(n-1)}P^{(n-2)}\cdots P^{(2)}P^{(1)}$$

也是正交的. 注意到对任意的正交矩阵 Q 总有 $Q^{-1} = Q^{T}$，所以得到 A 的 QR 分解(1-10)式.

1.3.2　求解线性方程组的QR算法

考虑线性方程组 $Ax = b$. 假设系数矩阵 A 有 QR 分解(1-10)，则有

$$QRx = b.$$

注意到 $Q^{T}Q = I$，所以得知

$$Rx = Q^{T}b.$$

而这是一个易于求解的上三角方程组.

1.4 经典迭代法及其收敛性

考虑线性代数方程组

$$Ax = b, \quad x, b \in \mathbf{R}^n. \tag{1-11}$$

它总可以等价地写成

$$x = Bx + f, \quad x, f \in \mathbf{R}^n. \tag{1-12}$$

给定初始向量 $x^{(0)} \in \mathbf{R}^n$,即可以构造迭代法

$$x^{(k+1)} = Bx^{(k)} + f, \qquad k = 0, 1, 2, \cdots, \tag{1-13}$$

其中的 $B \in \mathbf{R}^{n\times n}$ 称为**迭代矩阵**.

如果迭代公式(1-13)生成的序列 $\{x^{(k)}\}$ 有极限,则称**迭代法收敛**.假设

$$\lim_{k\to\infty} x^{(k)} = x^{(*)},$$

则 x^* 满足方程组(1-12).由于(1-12)式与(1-11)式是等价的,所以 x^* 满足原方程组(1-11).迭代公式(1-13)收敛的充分必要条件是

$$\rho(B) < 1.$$

对任意的算子范数(即可以由某种向量范数导出的矩阵范数),都一定有

$$\rho(B) \leqslant \| B \|,$$

所以 $\| B \| < 1$ 是迭代公式(1-13)收敛的充分条件,而且

$$\| x^{(k)} - x^{(*)} \| \leqslant \frac{\| B \|^k}{1 - \| B \|} \| x^{(1)} - x^{(0)} \|, \tag{1-14}$$

$$\| x^{(k)} - x^{(*)} \| \leqslant \frac{\| B \|}{1 - \| B \|} \| x^{(k)} - x^{(k-1)} \|. \tag{1-15}$$

(1-14)式给出的是迭代法收敛速度的一个估计.$\| B \|$ 越接近零,$\{x^{(k)}\}$ 收敛得越快;(1-15)式则可以作为停止计算的准则之一,它表明 $\| B \| < 1$ 但不很接近1时,只要相邻两次迭代 $x^{(k)}$、$x^{(k-1)}$ 足够靠近,则 $x^{(k)}$ 已经是 x^* 很好的近似了.

1.4.1 雅可比方法

取迭代矩阵

$$B = I - D^{-1}A, \quad f = D^{-1}b,$$

便得到雅可比(Jacobi)方法,其中

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad D = \begin{bmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & \ddots & \\ 0 & & & a_{nn} \end{bmatrix}.$$

雅可比方法的分量形式为

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\Big(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^{(k)}\Big), \quad i = 1, 2, \cdots, n. \tag{1-16}$$

所以,对某个问题,雅可比方法收敛的充要条件是

$$\rho(I - D^{-1}A) < 1.$$

1.4.2 高斯 - 塞德尔方法

在雅可比方法中,如果每步迭代是按 $x_1^{(k)} \to x_2^{(k)} \to \cdots \to x_n^{(k)}$ 的顺序计算的,则更新第 i 个分量时,前 $i-1$ 个分量已被更新过了,所以(1-16)式可以改进为

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x^{(k)}\right).$$

这便得到**高斯 - 塞德尔(Seidel)方法**.将它写成(1-13)式的形式即为

$$x^{(k+1)} = (D - L)^{-1}Ux^{(k)} + (D - L)^{-1}b.$$

其中

$$-L = \begin{bmatrix} 0 & & & 0 \\ a_{21} & 0 & & \\ a_{31} & a_{32} & \ddots & \\ \vdots & \vdots & & \ddots \\ a_{n1} & a_{n2} & & 0 \end{bmatrix}, \quad -U = \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & \ddots & & \vdots \\ 0 & & & \ddots & \vdots \\ & & & & 0 \end{bmatrix}.$$

对角占优矩阵是实际问题中常见的矩阵.设 $A = [a_{ij}]_{n\times n}$,如果

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1,2,\cdots,n,$$

则称 A 是严格对角占优的.系数矩阵为严格对角占优的线性代数方程组,它的雅可比方法和高斯 - 塞德尔方法都是收敛的.

1.4.3 超松弛方法

假设 $x^{(k)}$ 是第 k 步的迭代值,超松弛方法以 $x^{(k)}$ 为初值进行一步高斯 - 塞德尔迭代,得

$$\tilde{x}_i^{(k-1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1,2,\cdots,n.$$

第 $k+1$ 步的超松弛迭代取为

$$x_i^{(k+1)} = \omega\,\tilde{x}_i^{(k+1)} + (1-\omega)x_i^{(k)}, \tag{1-17}$$

其中 ω 称为**松弛参数**.将它写成矩阵形式为

$$x^{(k+1)} = (D - \omega L)^{-1}(\omega U + (1-\omega)D)x^{(k)} + (D - \omega L)^{-1}b.$$

这时它的迭代矩阵为

$$B_\omega = (D - \omega L)^{-1}(\omega U + (1-\omega)D).$$

当 $\omega = 1$ 时(1-17)式就是高斯 - 塞德尔方法,所以超松弛方法(亦称 SOR 方法)是高斯 - 塞德尔方法的加速与推广.超松弛方法的关键是如何选择使(1-17)式收敛得更快的松弛参数 ω.

使得超松弛方法收敛,ω 必须满足

$$0 < \omega < 2.$$

对满足特殊性质的系数矩阵,可以分析最佳松弛因子的选取问题,即寻找使 B_ω 的谱半径达到最小的松弛因子 ω.

1.4.4 分块迭代法

在实际问题中,所要求解的问题常常有自然的分块形式,如分块带状等.设

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} \end{bmatrix},$$

其中 A_{kk} 是 n_k 阶的方阵,而且 $n_1 + n_2 + \cdots + n_m = n$.记

$$A = D_B - L_B - U_B,$$

其中

$$D = \mathrm{diag}(A_{11}, A_{22}, \cdots, A_{mm}),$$

$$L_B = \begin{bmatrix} 0 & & & \\ A_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ A_{m1} & \cdots & A_{m,m-1} & 0 \end{bmatrix}, \quad U_B = \begin{bmatrix} 0 & A_{12} & \cdots & A_{1m} \\ & 0 & \ddots & \vdots \\ & & \ddots & A_{m-1,m} \\ & & & 0 \end{bmatrix}.$$

且 x_k、$b_k \in \mathbf{R}^{n_k}, k = 1,2,\cdots,m$,则求解线性代数方程组 $Ax = b$ 的分块雅可比迭代为

$$A_{ii}x_i^{(k+1)} = b_i - \sum_{\substack{j=1 \\ j \neq i}}^{m} A_{ij}\, x_j^{(k)}, \quad i = 1,2,\cdots,m.$$

分块的高斯 - 塞德尔迭代为

$$A_{ii}\, x_i^{(k+1)} = b_i - \sum_{j=1}^{i-1} A_{ij}\, x_j^{(k+1)} - \sum_{j=i+1}^{m} A_{ij}\, x_j^{(k)}{}_{ij}, \quad i = 1,2,\cdots,m.$$

类似地可以给出分块超松弛(BSOR) 方法.

1.5 矩阵求逆与行列式求值

这两者通常都是矩阵分解的副产品.

1.5.1 矩阵求逆

假设 A 是可逆的,记 A^{-1} 如下按列分块:

$$A^{-1} = (q_1, q_2, \cdots, q_n).$$

由于 $AA^{-1} = I$,则有

$$Aq_i = e_i, \quad i = 1,2,\cdots,n, \tag{1-18}$$

其中 e_i 为第 i 个分量为 1 其余分量为零的 n 维单位向量.利用 A 的 LU 分解或 QR

分解都可以较容易地解出(1-18)式中的 $q_i, i = 1,2,\cdots,n$.

应当指出的是,计算逆矩阵的计算代价是比较高的,除非特别必要外,应当尽可能避免.

1.5.2 行列式求值

假设矩阵 A 有 LU 分解:

$$A = LU,$$

从而有

$$\det(A) = \det(L)\det(U).$$

由于 L 是单位下三角的,所以 $\det(L) = 1$.而 U 是上三角的,它的行列式值只是其对角元素的乘积.

如果是在求解线性方程组的同时计算系数矩阵的行列式值,则其代价很小.例如用高斯列主元消去法求解方程组,它等价于

$$PA = LU,$$

P 的行列式很易求得,所以很易于求得 A 的行列式.

对一些特殊问题,如对称正定矩阵,有一些更有效的特殊算法.

2 求解大型稀疏问题的直接法

计算机的处理能力和运算速度在不断增长,但与此同时矩阵规模的增长似乎更快.20世纪60年代末阶数为10 000的线性方程组已属相当大规模,而今天它已是一般通用软件很容易处理的问题.**稀疏矩阵**是指其大部分的元素均为零的矩阵.如何具体描述“零元素很多”,则有许多不同的定义,如“非零元素不超过5 %”,“每行上的非零元素不超过10个”,“当 n 很大时,非零元素的个数与 n 同阶”等等.但这些都是比较表面化的.这一概念的根本点应当是:当值得通过考虑大量存在的零元素而获得明显利益时,这样的矩阵就是稀疏的.这里的基本因素有三个:矩阵、算法和所使用的计算机.任何稀疏的矩阵可以作为满的矩阵来处理,而任何满的矩阵也可以看成是稀疏的,只不过代价会更高(计算时间更长或需要的存储空间更大).本章集中讨论大型稀疏问题的直接方法(即消去法,深入的讨论见文献[3]和[4].),迭代法放到下一章.

2.1 带状矩阵的消去法

设 n 阶方阵 $A = [a_{ij}]_{n\times n}$,如果存在正整数 $m(m < n)$,使得

$$a_{ij} = 0, \quad 如果 \quad |i-j| > m,$$

则称 A 是**带状矩阵**,m 称为A 的**半带宽**.带状矩阵当 m 比较小时是稀疏的,而且其非零元素的分布具有极好的规律性,所以它也是最简单、最易于处理的稀疏矩阵,在实际问题中十分常见.特别当 $m = 1$ 时,A 的形状为

$$A=\begin{bmatrix}\alpha_1 & \gamma_1 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ & & \beta_n & \alpha_n\end{bmatrix},$$

它被称为三对角矩阵.

2.1.1　解三对角方程组的追赶法

不妨假设 $\beta_{i+1}\gamma_i\neq 0, i=1,2,\cdots,n-1$. 因为如果有某个 β_i 或 γ_i 为零，这时线性方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 等价于两个阶数更低的三对角方程组. 如果 $\boldsymbol{A}$ 非奇异且 $\boldsymbol{A}$ 的各阶顺序主子式非奇异，则有

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U},$$

其中

$$\boldsymbol{L}=\begin{bmatrix}1 & & & \\ l_{21} & 1 & & \\ & \ddots & \ddots & \\ & & l_{n,n-1} & 1\end{bmatrix},\quad \boldsymbol{U}=\begin{bmatrix}u_{11} & u_{12} & & \\ & u_{22} & \ddots & \\ & & \ddots & u_{n-1,n} \\ & & & u_{nn}\end{bmatrix},$$

而这里的

$$\begin{aligned}&u_{11}=\alpha_1, && l_{i,i-1}=\beta_i/u_{i-1,i-1},\\ &u_{i-1,i}=\gamma_{i-1}, && u_{ii}=\gamma_i-l_{i,i-1}u_{i-1,i},\end{aligned}\qquad i=2,3,\cdots,n.$$

这样方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的求解，等价于依次求解

$$\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b},\quad \boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}.$$

它们的解则依次为

$$\begin{aligned}&y_1=b_1,\\ &y_k=b_k-l_{k,k-1}y_{k-1},\quad k=2,\cdots,n;\\ &x_n=y_n/u_{nn},\\ &x_k=(y_k-u_{k,k+1}x_{k+1})/u_{kk},\quad k=n-1,\cdots,1.\end{aligned}$$

上述求解 $\boldsymbol{y}$ 的过程称为“追”，而计算 $\boldsymbol{x}$ 的过程称为“赶”，故称追赶法. 不难看出用追赶法求解三对角方程组计算量很小.

注意，如果矩阵 $\boldsymbol{A}$ 满足对角占优性质，即

$$\begin{aligned}&|\alpha_1|>|\gamma_1|,\\ &|\alpha_i|\geqslant|\beta_i|+|\gamma_i|>0,\quad i=2,3,\cdots,n-1,\\ &|\alpha_n|>|\beta_n|,\end{aligned}$$

则 $\boldsymbol{A}$ 非奇异且 $\boldsymbol{A}$ 的各阶顺序主子式都非奇异. 这时追赶法不但速度快，数值稳定性也好.

2.1.2　一般带状矩阵的消去法

假设线性代数方程组的系数矩阵

$$A=\begin{bmatrix} a_{11} & \cdots & a_{1,m+1} & \cdots & & \\ \vdots & \ddots & & \ddots & & \\ a_{m+1,1} & & \ddots & & \ddots & \\ \vdots & & & \ddots & & a_{n-m,n} \\ & & & & \ddots & \vdots \\ & & \cdots & a_{n,n-m} & \cdots & a_{nn} \end{bmatrix}$$

为带状矩阵，其半带宽为 $m(m \ll n)$. A 可以有如下带状特性的分解：

$$A = LU$$

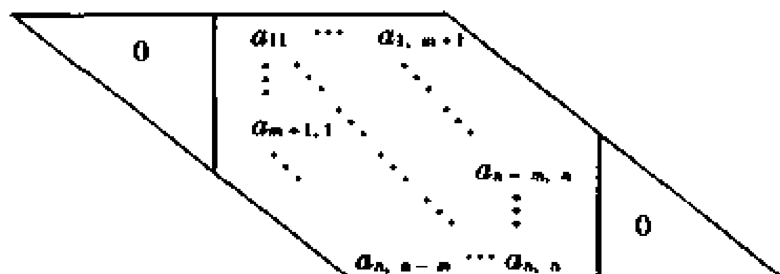
其中

$$L=\begin{bmatrix} 1 & & & & & \\ l_{21} & 1 & & & & \\ \vdots & \ddots & \ddots & & & \\ l_{m+1,1} & & \ddots & \ddots & & \\ & \ddots & & \ddots & \ddots & \\ & & l_{n,n-m} & \cdots & l_{n,n-1} & 1 \end{bmatrix}, U=\begin{bmatrix} u_{11} & \cdots & u_{1,m+1} & & & \\ & \ddots & & \ddots & & \\ & & \ddots & & \ddots & \\ & & & \ddots & & u_{n-m,n} \\ & & & & \ddots & \vdots \\ & & & & & u_{nn} \end{bmatrix}$$

应当特别注意的是，上面的讨论都是在可以顺序消元的前提下进行的.幸好实际问题中的带状矩阵，许多同时又是对称正定的，这时顺序消元的高斯消去法是稳定的.但如果采用主元素消去法，则必须同时考虑矩阵 A 的存储方式，这也是一般稀疏问题算法的重要方面.

2.1.3　带状矩阵的存储及主元消去法

针对三对角矩阵的追赶法，我们只须将三条对角线存成三个一维数组.对一般的带状矩阵，如果使用的是顺序高斯消去法，由于运算中不在带外产生非零元素，故可以将矩阵存为一个 n 行 $2m+1$ 列的数组，其中 m 为 A 的半带宽，使得每一列存放 A 的一个对角线.即存储如下的一个平行四边形：



对主元消去法，上述存储方式显然不再适用，全主元消去法可能会完全破坏带状矩阵的稀疏结构.注意到在上面所存的平行四边形中还有一些毫无用处的零元素，更好地利用这些空间可以构造出带状矩阵的列主元消去法(但不是 LU 分解).

下面以一个五阶三对角问题为例，阐述算法的基本思想.设有方程

$$Ax = b,$$

其中

$$A = \begin{bmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & a_{34} & \\ & & a_{43} & a_{44} & a_{45} \\ & & & a_{54} & a_{55} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}.$$

将 A 存成如下形式的数组

$$C: \begin{array}{ccc:c} a_{11} & a_{12} & 0 & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{32} & a_{33} & a_{34} & b_3 \\ a_{43} & a_{44} & a_{45} & b_4 \\ a_{54} & a_{55} & 0 & b_5 \end{array}$$

也就是说将 A 按行存，注意此时 $m = 1$，每步消元涉及的是 $m+1=2$ 行.

第一步消元，是在前 $m+1$ 行的第 1 列选主元，记主元为 $\overline{a}_{11}$，将主元所在行的其它元素除以 $\overline{a}_{11}$ 后记成 $\tilde{a}_{12}$、$\tilde{a}_{13}$，并将它们与原数组的第 1 行对换. 用第 1 行消去第 2 到第 $m+1$ 行的第 1 列元素. 注意 C 的第 2 行第 1 列元素被消成零，故将第 2 行(一般 $m \neq 1$ 的情况即第 2 到第 $m+1$ 行)的元素依次向前移，最后一个元素补零. 即此时

$$C: \begin{array}{ccc:c} \overline{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \hat{a}_{22} & \hat{a}_{23} & 0 & \hat{b}_2 \\ a_{32} & a_{33} & a_{34} & b_3 \\ a_{43} & a_{44} & a_{45} & b_4 \\ a_{54} & a_{55} & 0 & b_5 \end{array}$$

第 2 和第 3 步的消元结果依次为

$$\begin{array}{ccc:c} \overline{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \overline{a}_{22} & \tilde{a}_{23} & \tilde{a}_{24} & \tilde{b}_2 \\ \hat{a}_{33} & \hat{a}_{34} & 0 & \hat{b}_3 \\ a_{43} & a_{44} & a_{45} & b_4 \\ a_{54} & a_{55} & 0 & b_5 \end{array} \longrightarrow \begin{array}{ccc:c} \overline{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \overline{a}_{22} & \tilde{a}_{23} & \tilde{a}_{24} & \tilde{b}_2 \\ \overline{a}_{33} & \tilde{a}_{34} & \tilde{a}_{35} & \tilde{b}_3 \\ \hat{a}_{44} & \hat{a}_{45} & 0 & \hat{b}_4 \\ a_{54} & a_{55} & 0 & b_5 \end{array}$$

最终得到

$$\begin{array}{ccc:c}
\bar{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\
\bar{a}_{22} & \tilde{a}_{23} & \tilde{a}_{24} & \tilde{b}_2 \\
\bar{a}_{33} & \tilde{a}_{34} & \tilde{a}_{35} & \tilde{b}_3 \\
\bar{a}_{44} & \tilde{a}_{45} & 0 & \tilde{b}_4 \\
\hat{a}_{55} & 0 & 0 & \hat{b}_5
\end{array}$$

该数组的第1列是所有主元素,如果任何一个为零,消去过程应立刻停止.如果上述消去过程不中断,则得到一个与原方程等价的问题:

$$\begin{bmatrix} 1 & \tilde{a}_{12} & \tilde{a}_{13} & & \\ & 1 & \tilde{a}_{23} & \tilde{a}_{24} & \\ & & 1 & \tilde{a}_{34} & \tilde{a}_{35} \\ & & & 1 & \tilde{a}_{45} \\ & & & & \hat{a}_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \\ \tilde{b}_4 \\ \tilde{b}_5 \end{bmatrix},$$

回代便得到原问题的解.

2.2 稀疏矩阵的存储

从前面的讨论已经看到,稀疏矩阵的存储方式是构造稀疏方程组高效解法的关键.它应遵循如下的原则:

1° 尽可能少存或不存零元素;

2° 尽可能方便地找到非零元素在矩阵中的位置;

3° 尽可能便捷地填入新的非零元素和删除计算中产生的零元素.

上面的要求往往是相互矛盾的,选择怎样的存储方式与所选择的算法密切相关.下面介绍几种常用的方法.

2.2.1 逻辑尺方法

用$\lceil n^2/p \rceil + 1$个存储单元中的n^2个二进制位构成逻辑尺,其中p是计算机字长.上述n^2个二进制位依次为相应矩阵A的元素.如果矩阵中元素非零,则该二进制位取值为1,否则取0值,同时将A中的非零元素按与逻辑尺相同的次序(如按列)存入一个一维数组.当A的规模很大且稀疏程度很高时,这种方法效率并不高.该方法的优点是简单直观.

2.2.2 一维数组方法

这种方法需要三个一维数组

$$VE[1:\tau],RI[1:\tau],CI[1:n]$$

其中 n 是原稀疏矩阵 A 的阶数，τ 是 A 中非零元素的个数，通常 $n<\tau\ll n^2$. 数组 VE 按列存放 A 的非零元素；RI 存放 VE 中相应元素在 A 中的行号；CI 则依次存放 A 中各列第 1 个非零元素在 VE 中的位置.

这里只存了 A 的非零元素，要确定 A 在 (i,j) 位置上的元素也很容易，只要取

$$t_1 = CI(j),$$

$$t_2 = \begin{cases} CI(j+1)-1, & j\neq n, \\ \tau, & j=n. \end{cases}$$

如果有 $t_1\leqslant t\leqslant t_2$，使得

$$RI(t) = i,$$

则 $a_{ij}=VE(t)$，否则 $a_{ij}=0$. 但增加或删除元素却不易实现.

2.2.3　链表法

将 A 中每一个非零元素存为一个记录，它包含三个连续的单元，其内容分别是

元素所在的行
元素本身
本列下一个非零元素对应记录的首地址①

为更直观，下面看一个简单的例子，

$$\begin{bmatrix} 0 & 0 & a_{13} & 0 \\ a_{21} & 0 & 0 & a_{24} \\ 0 & a_{32} & a_{33} & 0 \\ a_{41} & 0 & 0 & 0 \end{bmatrix}$$

则它存为如下 6 个记录：

地址	101	201	910	485	980	620
行	2	4	3	1	3	3
元素	a_{21}	a_{41}	a_{32}	a_{13}	a_{33}	a_{24}
下一地址	201	0	0	980	0	0

① 如果该元素是同列中最后一个非零元素，则此处取值零.

每个记录框外的数字代表该记录的首地址，是随意假定的. 此外需要一个索引数组给出每列第一个非零元素的首地址. 对上面的例子就应当有

101	910	485	620

所以链表法共需 $3\tau + n$ 个存储单元.

链表法的优点是各记录可以不连续存放，可以充分利用零散的存储空间，而且填加和删除元素都较容易；缺点是程序实现上要较前几种方法困难.

2.3 随机稀疏矩阵的高斯消去法

高斯消去法虽然是一种很古老的方法，但至今仍是求解线性方程组的有效算法，许多成熟的数学软件仍以它作为核心的算法. 对于一个稀疏矩阵，消去法的每一步都会消去一些非零的元素，但通常还会同时产生一些新的非零元素. 新产生的非零元素称为填充(fill-in)，填充的数目称为填充量. 当用消去法求解稀疏的方程组时，为了不破坏矩阵的稀疏性，控制消去法的填充量是十分重要的. 除前面讨论过的带状矩阵外，诸如变带宽的带状矩阵、分块三对角矩阵、加边带状和加边块对角矩阵等，具有特殊稀疏结构的问题，也可以讨论它们特殊的消去方法. 这里仅介绍一般稀疏矩阵消去法的基本思想.

假设 A 是已知的 n 阶矩阵，定义 $\boldsymbol{B} = [b_{ij}]_{n\times m}$，满足

$$b_{ij} = \begin{cases} 0, & a_{i,j} = 0, \\ 1, & a_{i,j} \neq 0. \end{cases}$$

称 $\boldsymbol{B}$ 为 $\boldsymbol{A}$ 相应的布尔矩阵.

设 α 为一个布尔量，即它取值为0或为1，则 $\overline{\alpha}$ 为布尔代数中的“补”运算，其含义为：若 $\alpha = 0$，则 $\overline{\alpha} = 1$；若 $\alpha = 1$，则 $\overline{\alpha} = 0$. 由此我们可以定义上述布尔矩阵 $\boldsymbol{B}$ 的补矩阵为 $\overline{\boldsymbol{B}} = [\overline{b}_{i,j}]$.

2.3.1 局部填充量

设有线性方程组 $\boldsymbol{Ax} = \boldsymbol{b}$，其中 $\boldsymbol{A} = [a_{ij}]$. 若用 $a_{ij} \neq 0$ 作为主元时，这一步消去产生的填充量记为 g_{ij}，称为局部填充量. 利用前面定义的相应于 $\boldsymbol{A}$ 的布尔矩阵 $\boldsymbol{B}$ 和它的补，可以证明

$$\boldsymbol{G} = \boldsymbol{B}\overline{\boldsymbol{B}}^{\mathrm{T}}\boldsymbol{B},$$

其中 $\boldsymbol{G} = [g_{ij}]$. 也就是说用 $a_{ij} \neq 0$ 作主元时，这一步上的填充量可以计算出来. 所以，从保持稀疏性、减少填充量的角度选择主元，应取 $a_{\alpha\beta} \neq 0$. 使 α、β 满足

$$g_{\alpha\beta} = \min_{1\leq i,j\leq n} g_{ij}.$$

应当指出的是：选择上述的 $a_{\alpha\beta}$ 作主元，只是在这一步上填充量达到了极小，但从总体上看它不一定是最佳的. 所以这种原则产生的主元是局部最佳的. 另一方面，这样的原则完全是从节省内存的角度出发的，而完全没有考虑到舍入误差的作用. 在实际中找真正要用的主元，通常既不是填充量最小的，也不可能是舍入误差影响最小的，而需在两者间寻求一种平衡.

2.3.2　填充量的估计

上面给出了局部填充量 g_{ij} 的确切表达式，但由于在实际应用中必须同时考虑舍入误差的影响，也就是说所选主元 $a_{\alpha\beta}$ 不仅局部填充量 $g_{\alpha\beta}$ 要比较小，而且 $|a_{\alpha\beta}|$ 还要相对比较大.所以用比较小的计算量给出 g_{ij} 的估计是有意义的.

设 $a_{ij}\neq 0$，则用它作主元时的填充量 g_{ij} 满足如下关系：

$$g_{ij}\leqslant \tilde{g}_{ij}=((B-I)M(B-I))_{i,j},$$

其中 B 如上节所定义的，M 是全部元素都为 1 的 n 阶方阵.对任意矩阵 C，$(C)_{i,j}$ 表示该矩阵 (i,j) 位置上的元素.

3　求解大型稀疏问题的迭代法

迭代法在大型稀疏问题的研究中占有重要地位，本章介绍其中较成功的几种方法，详细的讨论请参见文献[1,2,4,5].

3.1　共轭梯度法

考虑线性代数方程组

$$Ax=b,\tag{3-1}$$

其中 A 是 n 阶对称正定矩阵，$b\in\mathbf{R}^n$.当 n 很大时任何直接法均告无效，必须借助迭代法.

3.1.1　变分原理

设 A 是 n 阶对称正定矩阵，$b\in\mathbf{R}^n$ 是已知向量.则 $x^*\in\mathbf{R}^n$ 是方程组(3-1)的解的充分必要条件是 x^* 满足

$$\varphi(x^*)=\min_{x\in\mathbf{R}^n}\varphi(x),$$

其中

$$\varphi(x)=\frac{1}{2}(x,Ax)-(x,b).$$

上述结果将对称正定线性代数方程组的问题等价为一个 n 元二次函数(亦称泛函)的极小问题.这一结果称为**变分原理或里兹(Ritz)原理**，是构造本节算法的基础.

3.1.2　最速下降法

由变分原理，计算 $\varphi(x)$ 极小的算法也就给出了求解方程组(3-1)的算法，而求得 $\varphi(x)$ 极小的关键当然是如何使 $\varphi(x)$ 的值下降.

设 x_0 是 $\mathbf{R}^n$ 中任意的一个点，则 $\varphi(x)$ 在该点下降最快的方向是它的负梯度方向.经计算知

$$-\operatorname{grad}\varphi(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}_0}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0=\boldsymbol{r}_0,$$

其中 $\boldsymbol{r}_0$ 称为残差向量.令

$$\boldsymbol{x}_1=\boldsymbol{x}_0+\alpha\boldsymbol{r}_0, \tag{3-2}$$

其中 α 待定.由于 $\varphi(\boldsymbol{x})$ 沿 $\boldsymbol{r}_0$ 的方向是下降的,可以期望

$$\varphi(\boldsymbol{x}_1)<\varphi(\boldsymbol{x}_0).$$

为使下降的效果最显著,自然的想法是令

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\varphi(\boldsymbol{x}_0+\alpha\boldsymbol{r}_0)=0. \tag{3-3}$$

这样选择的 α,使 $\boldsymbol{x}_1$ 沿 $\boldsymbol{r}_0$ 方向下降最多.不难给出(3-3)式的解是

$$\alpha_1=\frac{(\boldsymbol{r}_0,\boldsymbol{r}_0)}{(\boldsymbol{A}\boldsymbol{r}_0,\boldsymbol{r}_0)}.$$

如果在(3-2)式中取 $\alpha=\alpha_1$,就得到一步最速下降法.一般地

$$\boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\alpha_k\boldsymbol{r}_{k-1},\quad k=1,2,\cdots,$$

其中 $\boldsymbol{r}_{k-1}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_{k-1},\alpha_k=\dfrac{(\boldsymbol{r}_{k-1},\boldsymbol{r}_{k-1})}{(\boldsymbol{A}\boldsymbol{r}_{k-1},\boldsymbol{r}_{k-1})}$.

由于 $\boldsymbol{A}$ 是对称正定的,所以它所有的特征值都是正数,设为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n>0$.这时 $\|\boldsymbol{x}\|_A=(\boldsymbol{x},\boldsymbol{A}\boldsymbol{x})^{1/2}$ 给出的是一种向量范数.最速下降法的收敛性有如下的结果:

$$\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_A\leqslant\left(\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}\right)^k\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_A,$$

其中 $\boldsymbol{x}^*$ 是方程组(3-1)的精确解,$\boldsymbol{x}_0$ 为初始点.

上述结果表明最速下降法是收敛的,收敛的速度取决于 A 的特征值分布的情况.如果 $\lambda_1\gg\lambda_n$,则收敛是相当慢的.

还应当注意的是,$\boldsymbol{r}_k$ 是 $\varphi(\boldsymbol{x})$ 在 $\boldsymbol{x}_k$ 处的最速下降方向,但这是局部的.以尽快到达 $\varphi(\boldsymbol{x})$ 的极小为目标,如此的选择不一定是最优的.另一方面,特别当 $\boldsymbol{r}_k$ 比较小时,实际计算中由于舍入误差的影响会偏离最速下降方向,从而计算中显示函数值不稳定性.

最速下降法并不是很实用的方法,但它是众多新算法研究的出发点.

3.1.3 共轭梯度法

最速下降法中对 $\varphi(\boldsymbol{x})$ 的每步极小化都是局部的,共轭梯度法的基本思想是对 $\varphi(\boldsymbol{x})$ 极小化过程要整体化.如果在 $\mathbf{R}^n$ 中能取到 $\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_k$ 是线性无关的,使 $\boldsymbol{x}_k$ 满足

$$\varphi(\boldsymbol{x}_k)=\min_{\boldsymbol{x}\in S_k}\varphi(\boldsymbol{x}), \tag{3-4}$$

其中 $S_k=\operatorname{span}\{\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_k\}$,即由 $\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_k$ 张成的线性空间,则 $k=n$ 时就给出了方程组(3-1)的解.

当 $k=1$ 时,只要取 $\boldsymbol{p}_1=\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$,就满足(3-4)式的要求.

假设有 $p_1,p_2,\cdots,p_{k-1}(k\geqslant 2)$ 使得

$$x_{k-1}=\sum_{i=1}^{k-1}\alpha_i p_i,$$

满足
$$\varphi(x_{k-1})=\min_{x\in S_{k-1}}\varphi(x).$$

共轭梯度法只要取

$$x_k=x_{k-1}+\alpha_k p_k.$$

这里要求选择 p_k,使得

$$(p_i,Ap_k)=0,\quad i=1,2,\cdots,k-1. \tag{3-5}$$

称 p_k 是与 $p_1,\cdots,p_{k-1}$ 是 A 共轭的,而

$$\alpha_k=(b,p_k)/(Ap_k,p_k).$$

满足(3-5)式的非零向量是可以构造出来的,即取

$$p_k=r_{k-1}+\beta_k p_{k-1},$$

其中
$$\beta_k=-\frac{(p_{k-1},Ar_{k-1})}{(p_{k-1},Ap_{k-1})}.$$

如此构造出 $\{x_k\}$ 就是求解方程组(3-1)的共轭梯度法.

从理论上讲,共轭梯度法是一种直接方法.由方法的构造过程可见,若 A 是 n 阶矩阵,则 x_n 应当是方程组(3-1)的精确解.但由于实际计算中舍入误差的影响,$\{p_i\}_{i=1}^k$ 不可能严格保持 A 正交的,甚至当 k 大到一度程度后,$\{p_i\}_{i=1}^k$ 可能变得"几乎线性相关",所以共轭梯度法更多是作为迭代法使用的.

若 A 是对称正定的,它的特征值设为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n>0$,则求解方程组(3-1)的共轭梯度法给出的向量序列满足如下估计:

$$\|x_k-x^*\|_A\leqslant 2\left(\frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}+\sqrt{\lambda_n}}\right)^k\|x_0-x^*\|_A.$$

可见共轭梯度法比最速下降法收敛速度更快.但当 $\lambda_1\gg\lambda_n$ 时,共轭梯度法的收敛还是比较慢的.以它为基础发展起来了一系列实用的计算方法.

3.1.4 预条件共轭梯度法

共轭梯度法作为一种实用的迭代法,其优点是:可充分利用系数矩阵的稀疏性,计算步骤简明,易于并行计算等.为克服 $\lambda_1\gg\lambda_n$ 时收敛太慢的缺点,预处理技术是必要的辅助手段,它的基本思想是将方程组(3-1)化成与之等价的方程组

$$\tilde{A}x=\tilde{b},$$

其中 $\tilde{A}$ 仍然是对称正定的,而且 $\tilde{A}$ 最大与最小特征值之比远远小于 A 的比值.

常用的预处理技术之一是多项式预处理,即选择一个次数不高的实系数多项式 $C(\lambda)$,则方程组(3-1)便等价于

$$C(A)Ax=C(A)b.$$

如果能取 $C(A)=A^{-1}$,即 $C(\lambda)=1/\lambda$,则经过预处理的方程已不用再求解.这当然不可能,况且我们要求 $C(\lambda)$ 是多项式.但我们自然可以试图求一个次数不高于

$m-1<n$ 的多项式，使

$$\max_{\lambda\in I}\left|\frac{1}{\lambda}-C(\lambda)\right|$$

尽可能小，其中 I 是包含了 A 全部特征值的一个区间. 它等价于求 $C_{m-1}(\lambda)$，使

$$\max_{\lambda\in S}|1-C_{m-1}(\lambda)\lambda|=\min_{C(\lambda)\in\pi_{m-1}}\max_{\lambda\in S}|1-C(\lambda)\lambda|,$$

其中 π_{m-1} 代表所有次数不高于 $m-1$ 次的实系数多项式的全体. 这是一个逼近论问题，$C_{m-1}(\lambda)$ 的表达式可以借助切比雪夫多项式给出.

3.2　不完全因子分解

如果 A 是对称正定的，则它一定有楚列斯基分解，但若 A 还是大型稀疏的，则分解很可能完全破坏 A 的稀疏性. 不完全楚列斯基分解，就是将 A 分解成

$$A=LL^{\mathrm{T}}+R,\tag{3-6}$$

其中 L 是下三角矩阵，R 称为**剩余矩阵**. 由于这里 R 是可以变化的，所以 L 的稀疏性可以预先规定，比如可以要求 L 与 A 有相同的稀疏性，从而克服楚列斯基分解破坏矩阵稀疏性的问题.

类似(3-6)式的不完全分解有相当大的灵活性，不仅可以指定 L 的结构，而且 R 中元素的选择也有相当大的余地. 不完全因子分解最成功的应用就是作为预处理手段.

3.2.1　不完全 LU 分解

设矩阵 $A=[a_{ij}]$ 为 n 阶的，

$$I_n=\{(i,j)\mid 1\leqslant i,j\leqslant n,i\neq j\}$$

称为指标集. 令

$$I_A=\{(i,j)\in I_n\mid a_{ij}\neq 0\},$$

也就是说 I_A 是 A 非零非对角元素的指标集.

对 I_n 的任意包含 I_A 的子集合 $\tilde{I}$ 及任一参数 $\omega,0\leqslant\omega\leqslant 1$，若 A 有分解

$$A=LU+R,\tag{3-7}$$

其中 $L=[l_{ij}]$ 是单位下三角矩阵，且

$$l_{ij}=0,\quad\forall(i,j)\in\tilde{I}\text{ 或 }i<j;$$

$U=[u_{ij}]$ 是上三角矩阵，且

$$u_{ij}=0,\quad\forall(i,j)\in\tilde{I}\text{ 或 }i>j;$$

而剩余矩阵 $R=[r_{ij}]$ 满足

$$r_{ij}=0,\qquad\forall(i,j)\in\tilde{I},$$
$$r_{ii}=-\omega\sum_{j\neq i}r_{ij},\quad i=1,2,\cdots,n,$$

则称(3-7)式为 A 关于 $\tilde{I}$ 和 ω 的**松弛不完全 LU 分解**，ω 称为**松弛参数**.

在上述分解中，$\omega = 0$对应著名的不完全LU分解；$\omega = 1$即是修正不完全LU分解.分解(3-7)式可以由高斯消去的修正得到.很多实际中常见的矩阵，其松弛不完全LU分解是存在的.在实际计算中，可将L存放在A的下三角部分原占用的空间中，将U存放在A的上三角部分.通常R不需要，故不保存它.

3.2.2　不完全楚列斯基分解

现假设A是对称正定的，这时A的指标集$\tilde{I}$也是对称的，即若有$(i,j) \in \tilde{I}$，则有$(j,i) \in \tilde{I}$.对应于一般矩阵的松弛不完全LU分解，可以考虑对任意$I_A \subset \tilde{I} \subset I_n$及$0 \leqslant \omega \leqslant 1$，$A$关于$\tilde{I}$和$\omega$的**松弛不完全楚列斯基分解**：

$$A = LDL^{\mathrm{T}} + R, \tag{3-8}$$

其中D是对角矩阵，L是下三角矩阵且

$$l_{ij} = 0, \quad \forall (i,j) \in I, \text{或 } i < j.$$

易知这里的剩余矩阵R也是对称的.对很多实际常见的矩阵，若A是对称正定的，则它的松弛不完全楚列斯基分解(3-8)式存在，而且LDL^{T}是正定的，即D的各对角元素都是正的.

3.3　GMRES算法

科学与工程计算中，大型线性代数方程组(3-1)是基本问题.多数情况下A是稀疏的，但不具有对称正定性.处理这类问题，本节介绍的基于伽辽金(Galerkin)原理的一类算法有重要价值.

3.3.1　伽辽金原理

取$x_0 \in \mathbf{R}^n$是任意向量，令$x = x_0 + z$，则方程组(3-1)等价于方程

$$Az = r_0, \tag{3-9}$$

其中$r_0 = b - Ax_0$.以下总使用方程组(3-9)的形式.

设K_m和L_m是$\mathbf{R}^n$中两个m维的子空间，它们分别由$\{v_i\}_{i=1}^m$和$\{w_i\}_{i=1}^m$张成.求解方程组(3-9)的**伽辽金原理**为：求$z_m \in K_m$作为方程组(3-9)的近似解，使

$$(r_0 - Az_m, w) = 0, \quad \forall w \in L_m. \tag{3-10}$$

令$V_m = (v_1, v_2, \cdots, v_m)$，$W_m = (w_1, w_2, \cdots, w_m)$，则它们都是$n \times m$的矩阵.由于$z_m \in K_m$，故有$y_m \in \mathbf{R}^m$，使得$z_m = V_m y_m$.所以(3-10)式可改写成

$$(W_m^{\mathrm{T}} A V_m) y_m = W_m^{\mathrm{T}} r_0. \tag{3-11}$$

求解方程组(3-11)可以给出方程组(3-9)解的近似.注意方程组(3-11)是一个m维的问题，只有当$m = n$时才能给出方程组(3-9)的精确解.

空间K_m和L_m不同的选取便给出不同的算法.当$m \ll n$时，我们是用一个m维问题(3-11)的解近似原来大规模问题(3-9)的解，当m增加时z_m是否能更靠近方

程组(3-9)的解呢?下面我们讨论一些具体的算法.

3.3.2　阿诺尔德过程

设 r_0 为给定的向量,则 $\mathrm{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$ 称为克雷洛夫(Krylov)子空间,它在线性方程组的近现代算法中占有重要地位.阿诺尔德(Arnold)过程是建立克雷洛夫(Krylov)子空间标准正交基的算法.

首先取

$$v_1 = \frac{r_0}{\|r_0\|_2}.$$

假设已有 $v_1, v_2, \cdots, v_k \in \mathrm{span}\{r_0, Ar_0, \cdots, A^{k-1}r_0\}$ 且它们是相互正交的,则 v_{k+1} 的构造过程如下:令

$$\tilde{v}_{k+1} = Av_k - \sum_{i=1}^{k} h_{i,k}v_i,$$

其中

$$h_{i,k} = (Av_k, v_i), \quad i = 1,2,\cdots,k.$$

则取

$$v_{k+1} = \tilde{v}_{k+1}/h_{k+1,k},$$

其中

$$h_{k+1,k} = \|\tilde{v}_{k-1}\|_2.$$

若 $m < n$,且 $h_{k+1,k} \neq 0, k = 1,2,\cdots,m-1$,则如上构造出的向量组 $\{v_i\}_{i=1}^{m}$ 构成 $\mathrm{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$ 的一组标准正交基,而且对所有的 $k < m$,都有 $v_{k+1} \perp \mathrm{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$.

阿诺尔德过程也可以用矩阵符号写成:

$$AV_k = V_kH_k + h_{k+1,k}v_{k+1}e_k^{\mathrm{T}}, \tag{3-12}$$

其中的 $V_k = (v_1, v_2, \cdots, v_k)^{\mathrm{T}}$,

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,k-1} & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2,k-1} & h_{2k} \\ 0 & h_{32} & \cdots & h_{3,k-1} & h_{3k} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & h_{k,k-1} & h_{kk} \end{bmatrix}.$$

3.3.3　阿诺尔德算法

若在伽辽金原理中取 $K_m = L_m = \mathrm{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$,伽辽金原理即是阿诺尔德算法.这时方程组(3-11)有如下的简单形式

$$H_my_m = \beta e_1, \tag{3-13}$$

其中 $\beta = \|r_0\|$, $e_1 = (1,0,\cdots,0)^{\mathrm{T}} \in \mathbf{R}^m$.如果 H_m 非奇异,方程组(3-13)由 $z_m = V_my_m$ 给出了方程组(3-9)解的一个近似;如果 H_m 奇异,则称算法出现恶性中断,将会无功而返.

关于阿诺尔德算法的残差有

$$\|r_0 - Az_m\|_2 = h_{m+1,m}\,|e_m^{\mathrm{T}}y_m|,$$

它是实际计算中判别 z_m 近似程度的重要尺度.

注意到当问题的规模 n 较大时,m 通常也会相当大,这时为了求得 z_m 需要保存 $\{v_i\}_{i=1}^m$,这使得算法存储的开销过大.而所谓循环型的算法是实际中常用的.其基本思想是:选定一个 $m \ll n$,执行了 m 步的阿诺尔德算法后,以第 m 步所得的残差 r_m 作为初始向量,重新启动阿诺尔德算法.

阿诺尔德算法的最大弱点是无法预料恶性中断何时发生.可以构造出这样的例子,不论 $m < n$ 如何选取,算法都会出现恶性中断.

另外,对一般的矩阵,阿诺尔德算法的收敛性至今仍没有给出证明.然而不少数值计算又表明,循环阿诺尔德算法是很有效的.

3.3.4 GMRES 算法

如果取 $K_m = \text{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$,而 $L_m = \text{span}\{Ar_0, A^2r_0, \cdots, A^m r_0\} = AK_m$,这时的伽辽金原理即给出 **GMRES 算法**(Generalized Minimal RESidual Algorithm),即广义极小残余算法,该算法已成为求解一般大型稀疏问题的重要工具.对空间 K_m 和 L_m 选取 $z_m = V_m y_m$,则 y_m 是方程组(3-11)解的充分必要条件是

$$\| r_m \|_2 = \| r_0 - Az_m \|_2 = \min_{z \in K_m} \| r_0 - Az \|_2. \tag{3-14}$$

这也是方法名称的来由,它将方程组(3-11)的求解等价为残余向量 2- 范数的极小化问题.GMRES 算法与阿诺尔德算法的差别在于,它每步构造 z_m,要求解一个最小二乘问题(3-14),而不是线性方程组(3-13).与阿诺尔德算法同样的理由,GMRES 算法也有循环的 GMRES 算法,它也是实际问题中常用的.

在通常情况下,求解线性方程组要比求解最小二乘问题简单,但这里的(3-14)式有很特殊的结构,所以有比较有效的解法.另一方面它也克服了阿诺尔德算法可能有恶性中断的困难.

对原问题附加一定的条件还可以证明 GMRES 算法的收敛性.例如,假设 A 是可以相似对角化的,即存在非奇异矩阵 X,使得 $A = X\Lambda X^{-1}$,其中 $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$,$\lambda_i, i = 1, 2, \cdots, n$,是 A 的特征值,则对充分大的 m,循环的 GMRES 算法是收敛的.

3.4 最小二乘问题

线性代数方程组讨论的是适定的线性问题,其方程的个数等于未知数的个数.设有

$$Ax = b, \tag{3-15}$$

如果 $A \in \mathbf{R}^{m \times n}$ 且 $m \neq n$,则上述方程或者无解($m > n$ 时,通常如此),或者不能唯一定解($m < n$ 时).所谓最小二乘问题,是指求 $x \in \mathbf{R}^n$,使

$$\| Ax - b \|_2 = \min_{v \in \mathbf{R}^n} \| Av - b \|_2. \tag{3-16}$$

而在所有满足(3-16)式的 x 中,范数最小的称为**最小范数解**.最小范数解是唯一的,(3-16)式有唯一解的充分必要条件是

$$\text{rank}(A) = n.$$

容易证明,x 是(3-16)式的解,当且仅当

$$A^{\mathrm{T}}Ax = A^{\mathrm{T}}b. \tag{3-17}$$

方程(3-17)称为**最小二乘问题的法方程**.

3.4.1 法方程方法

这是一个古老而直观的方法.如果 A 是满秩的(且假设 $m > n$),则方程(3-17)的系数矩阵 $A^{\mathrm{T}}A$ 是对称正定的.如果 n 不很大,可以利用 $C = A^{\mathrm{T}}A$ 的楚列斯基分解给出方程(3-17)的解,从而得到(3-16)式的解.当 $m \gg n$ 时,存储 $A^{\mathrm{T}}A$ 要比存储 A 更经济.

然而该方法一个显而易见的缺点是,方程(3-17)的条件数是 A 条件数的平方,特别当 A 已经比较病态时,这种方法的数值稳定性较差.

3.4.2 正交化方法

仍假设 $m > n$ 且 $\text{rank}(A) = n$.若 A 有 QR 分解,

$$A = Q\begin{bmatrix} R \\ O \end{bmatrix} = [Q_1 \quad Q_2]\begin{bmatrix} R \\ O \end{bmatrix} = Q_1 R.$$

其中 $Q_1 \in \mathbf{R}^{m\times n}, Q_2 \in \mathbf{R}^{m\times(m-n)}, R \in \mathbf{R}^{n\times n}$.则(3-16)式等价于

$$\| Q^{\mathrm{T}}(Ax - b) \|_2 = \min_{v\in\mathbf{R}^n} \| Q^{\mathrm{T}}(Av - b) \|_2.$$

如果记 $d_1 = Q_1^{\mathrm{T}}b$,只要求解三角方程

$$Rx = d_1,$$

就可以给出最小二乘问题(3-16)的解.

该方法具有良好的数值稳定性,通常计算结果比较准确,但计算的代价也比较高.

4 矩阵特征值问题

本章讨论矩阵特征值问题

$$Au = \lambda u,$$

的数值方法,并简单介绍广义特征值问题

$$Au = \lambda Bu,$$

其中 A、$B \in \mathbf{R}^{n\times n}$.文献[7]是这方面的经典著作;深入的讨论参见文献[4]和[6].

4.1 特征值问题的条件

先考虑两个极端的例子.设

$$A(\epsilon)=\begin{bmatrix}0 & & & \epsilon\\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0\end{bmatrix},\quad B(\epsilon)=\begin{bmatrix}0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ \epsilon & & 1 & 0\end{bmatrix}.$$

显然 $A(0)=B(0)$ 是严格下三角的矩阵,它所有的特征值都是 0. 若取矩阵的阶数为 20,取 ϵ 为 10^{-20},这时 $A(\epsilon)$ 特征值的模为 10^{-1},而 $B(\epsilon)$ 的特征值全是零. $A(\epsilon)$ 与 $B(\epsilon)$ 都是对 $A(0)$ 的扰动. 机器的存储所导致的舍入误差,足以使 $A(\epsilon)$ 形式的扰动引起矩阵特征值显著的变化,而 ϵ 再大也不会影响 $B(\epsilon)$ 的特征值. 这表明特征值可以对有些元素的变化敏感,而对另外一些元素不敏感;不同的特征值对不同的元素敏感性还可能不同. 由此可见,全面刻划特征值的敏感性是极其复杂的,只有在某些特殊条件下才能讨论.

假设 A 是可以对角化的,即有非奇异矩阵 P,使得

$$P^{-1}AP=\operatorname{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n).\tag{4-1}$$

设有矩阵范数 $\|\cdot\|$,它对任意的对角矩阵 $D=\operatorname{diag}(d_1,d_2,\cdots,d_n)$,都有

$$\|D\|=\max_{1\leq k\leq n}|d_k|,\tag{4-2}$$

(注意,常用的范数如 $\|\cdot\|_1,\|\cdot\|_2,\|\cdot\|_\infty$ 等都是满足上述条件的),则对任意矩阵 δA,都有

$$sp(A+\delta A)\subset\bigcup_{k=1}^{n}S_k,\tag{4-3}$$

其中的 $sp(M)$ 表示矩阵 M 特征值的集合,而

$$S_k=\{z\in\mathbb{C}\,|\,|z-\lambda_k|\leq\operatorname{cond}(P)\|\delta A\|\},$$

这里的 $\operatorname{cond}(P)=\|P\|\,\|P^{-1}\|$,即大家熟知的矩阵条件数.

注意,使(4-1)式成立的 P 并不是唯一的,而(4-3)式是对所有满足(4-1)式的 P 都成立的,若定义

$$\Gamma(A)=\inf_{P\in S}\{\operatorname{cond}(P)\},$$

其中 $S=\{P\in\mathbf{R}^{n\times n}\,|\,P^{-1}AP=\operatorname{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)\}$,则有

$$sp(A+\delta A)\subset\bigcup_{k=1}^{n}\{z\in\mathbb{C}\,|\,|z-\lambda_k|\leq\Gamma(A)\|\delta A\|\}.$$

所以也称 $\Gamma(A)$ 是矩阵 A 关于特征值问题的条件数.

应该特别强调的是,矩阵关于特征值问题的条件数和关于方程求解问题的条件数是完全不同的两个概念. 例如对

$$A=\begin{bmatrix}1 & 0\\ 0 & 10^{-6}\end{bmatrix},$$

它已是对角的,所以关于特征值问题的谱条件数 $\Gamma_2(A)=1$ 是良态的问题;而关于方程求解问题的谱条件数则为

$$\operatorname{cond}_2(A)=\|A\|_2\|A^{-1}\|_2=10^6,$$

对方程求解问题它是坏条件的.

4.2　幂法与反幂法

本节介绍特征值问题最简单、最古朴的几种方法，虽然计算特征值问题时已很少使用，但其中的思想是可以借鉴的.

4.2.1　幂法

取 $x_0 \in \mathbf{C}^n$，构造迭代

$$y_{k+1} = Ax_k,$$

$$x_{k+1} = \frac{y_{k+1}}{\max(y_{k+1})}, \quad k = 0,1,2,\cdots,$$

此即是所谓的幂法，其中 $\max(y)$ 代表 y 按模最大的分量. 设 A 的特征值满足

$$|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|.$$

如果存在 $P \in \mathbf{R}^{n\times n}$，使

$$P^{-1}AP = \mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n),$$

记 $P = (u_1,u_2,\cdots,u_n)$，则 $u_k \in \mathbf{C}^n$ 是对应于λ_k的特征向量. 故存在 $\alpha_1,\alpha_2,\cdots,\alpha_n \in \mathbf{C}$，使

$$y_0 = \sum_{i=1}^{n}\alpha_i u_i.$$

只要其中的 $\alpha_1 \neq 0$，则当 $k \to \infty$ 时有

$$x_k \to u_1, \quad \max(y_k) \to \lambda_1.$$

4.2.2　反幂法

取 $w_0 \in \mathbf{C}^n$，反幂法产生如下的序列：

$$Av_{k+1} = w_k,$$

$$w_{k+1} = \frac{v_{k+1}}{\max(v_{k+1})}, \quad k = 0,1,2,\cdots.$$

假设 A 是可以对角化的，且它的特征值满足

$$|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_{n-1}| > |\lambda_n|,$$

即最小的特征值是按模分离的，则存在 $\beta_1,\beta_2,\cdots,\beta_n$，使得

$$w_0 = \beta_1 u_1 + \beta_2 u_2 + \cdots + \beta_n u_n.$$

如果其中的 $\beta_n \neq 0$，则当 $k \to \infty$ 时，

$$w_k \to u_n, \quad \max(v_k) \to \lambda_n.$$

这表明幂法给出按模最大的特征值和特征向量，反幂法给出按模最小的特征值和特征向量.

4.2.3　带位移的反幂法

设 $\tilde{\lambda} \in sp(A)$，即 $\tilde{\lambda}$ 不是 A 的特征值. 又设 λ 是A的特征值，q 是相应的特征

向量,而且

$$|\tilde{\lambda}-\lambda|<|\tilde{\lambda}-\mu|, \quad \forall \mu \in sp(A) \setminus \{\lambda\},$$

也就是说 λ 是离 $\tilde{\lambda}$ 最近的特征值. 取

$$v_0 = q + \sum_{j=1}^{m} \gamma_j q_j,$$

其中 $q_1, q_2, \cdots, q_m$ 是相应于 A 的其它特征值 $\mu_1, \mu_2, \cdots, \mu_m$ 的特征向量. 迭代法

$$(A - \tilde{\lambda} I) \eta_{k+1} = v_k,$$

$$v_{k+1} = \frac{\eta_{k+1}}{\max \eta_k}, \quad k = 0,1,2,\cdots$$

称为带位移的反幂法. 当 $k \to \infty$ 时有

$$v_k \to q, \quad \max \eta_k \to \lambda.$$

注意到, $\tilde{\lambda}$ 靠近 λ 会使反幂法收敛得更快, 但这时矩阵 $A - \tilde{\lambda} I$ 更加病态, 从而会造成迭代过程上的困难. 这也就是说, 带位移的反幂法的收敛速度与算法的稳定性是对立的.

4.3 雅可比方法

假设 A 是对称的. 雅可比方法的基本思想是构造简单、特殊的正交矩阵序列 $\{P_k\}$, 使

$$A_0 = A,$$

$$A_{k+1} = P_k A_k P_k^{\mathrm{T}}, \quad k = 0,1,2,\cdots,$$

逐渐接近对角型, 从而得到 A 的特征值.

4.3.1 二维情形

先考虑 $n = 2$ 的简单情形. 设有矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad a_{21} = a_{12},$$

取矩阵

$$P = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1,$$

该矩阵确是正交的. 取 $c = \cos\theta_1, s = \sin\theta$,

$$C = PAP^{\mathrm{T}} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix},$$

$$c_{21} = c_{12}.$$

如果 $a_{21} = 0$, 则 A 已是对角的; 否则只要取

$$\text{ctan}2\theta = \frac{a_{11} - a_{22}}{2a_{21}}, \quad \theta \in \left(-\frac{\pi}{4}, 0\right) \cup \left(0, \frac{\pi}{4}\right),$$

它使 $c_{21} = c_{12} = 0$. 由于 $\boldsymbol{P}$ 是正交的,所以

$$C_{11}^2 + C_{22}^2 = a_{11}^2 + a_{22}^2 + 2a_{12}^2.$$

4.3.2 一般情形

对一般的 n 阶矩阵 $\boldsymbol{A} = [a_{ij}]_{n\times n}$,如果 $a_{ij} \neq 0$,令

$$\text{ctan}2\theta = \frac{a_{ii} - a_{jj}}{2a_{ij}}, \quad \theta \in \left(-\frac{\pi}{4}, 0\right) \cup \left(0, \frac{\pi}{4}\right).$$

取矩阵

$$\boldsymbol{P}(i,j) = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & s & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & -s & & & c & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \\ \end{matrix} \qquad (4\text{-}4)$$

$$\qquad\qquad\qquad i \qquad\qquad\quad j$$

则矩阵 $[c_{ij}]_{n\times n} = \boldsymbol{C} = \boldsymbol{PAP}^{\mathrm{T}}$ 中的元素 $c_{ij} = c_{ji} = 0$,而且有

$$\sum_{i,j=1}^{n} a_{ij}^2 = \sum_{i,j=1}^{n} c_{ij}^2.$$

注意,矩阵 $\boldsymbol{A}$ 变换到 $\boldsymbol{C}$,两个矩阵的差别仅在第 i、j 两行和 i、j 两列. 对任意的矩阵 $\boldsymbol{M} = [m_{ij}]_{n\times n}$,记

$$DS(\boldsymbol{M}) = \sum_{i=1}^{n} m_{ii}^2, \quad OS(\boldsymbol{M}) = \sum_{i\neq j} m_{ij}^2$$

分别表示对角和非对角元素的平方和,则有

$$DS(\boldsymbol{C}) = DS(\boldsymbol{A}) + 2(a_{ij})^2,$$

$$OS(\boldsymbol{C}) = OS(\boldsymbol{A}) - 2(a_{ij})^2.$$

上述结果表明,每次雅可比迭代,其效果是对角线元素"比份"增加,变化的多少取决于 a_{ij} 的大小. 雅可比方法的最终目的是使

$$\lim_{k\to\infty} OS(\boldsymbol{A}_k) = 0.$$

要强调指出的是,某个元素变成零,在后面的迭代中仍可能再变成非零. 选择 (i,j) 的原则,应使 a_{ij} 尽可能大.

4.4 QR 算法

节 1.3 介绍了矩阵的 QR 分解，本节基于此构造计算特征值的数值方法.

4.4.1 方法的原理

假设 A 是任意的方阵.记 $A_1 = A$,则 QR 算法的基本过程是:

对 $k = 1,2,\cdots$

(1) 计算 A_k 的 QR 分解:$A_k = Q_k R_k$.

(2) 构造新的矩阵:$A_{k+1} = R_k Q_k$.

上述过程给出了一个矩阵的序列$\{A_k\}$,而且

$$A_{k+1} = R_k Q_k = Q_k^{\mathrm{T}} A_k Q_k, \quad k = 1,2,\cdots.$$

这表明$\{A_k\}$中的每个矩阵都是相互正交相似的,从而都与 $A_1 = A$ 正交相似,所以该矩阵序列具有相同的特征值.

如果 A 是可逆的且它的特征值满足

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0,$$

再若 $A_k = [a_{ij}^{(k)}]_{n\times n}$,则通常情况下会有

$$\lim_{k\to\infty} a_{ii}^{(k)} = \lambda_i, \quad i = 1,2,\cdots,n,$$

$$\lim_{k\to\infty} a_{ij}^{(k)} = 0, \quad \forall\, i > j, \quad 1 \leqslant i,j \leqslant n.$$

对 $i < j$ 部分的元素,$k\to\infty$ 时它可以没有极限,也就是说 A_k 作为矩阵序列是可以不收敛的.

QR 算法的收敛性是数值代数中具有重要意义的结果.它的证明过程克服了分析中的重大困难.

当 A 有模相等的特征值时,当 $k\to\infty$,A_k 在结构上趋于分块上三角矩阵.这时对角块的矩阵不一定收敛,但它的特征值是收敛的.

4.4.2 海森伯格矩阵

这是一种特殊形式的矩阵,它在计算特征值的 QR 算法中具有重要意义.上海森伯格(Hessenberg)矩阵为

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2n} \\ & h_{32} & \cdots & \cdots & h_{3n} \\ & & \ddots & & \vdots \\ & & & h_{n,n-1} & h_{n,n} \end{bmatrix}. \tag{4-5}$$

它的转置矩阵也称下海森伯格矩阵.

设 A 是任意的方阵,记

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}=(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\cdots,\boldsymbol{\alpha}_n).$$

令 $\tilde{\boldsymbol{\alpha}}_1=(a_{21},a_{31},\cdots,a_{n1})^{\mathrm{T}}\in\mathbf{R}^{n-1}$,如果取

$$\boldsymbol{u}_1=\sigma_1(\tilde{\boldsymbol{\alpha}}_1-s_1\boldsymbol{e}_1)\in\mathbf{R}^{n-1},$$

其中 $\boldsymbol{e}_1=(1,0,\cdots,0)^{\mathrm{T}}\in\mathbf{R}^{n-1}$,且

$$s_1=-\|\tilde{\boldsymbol{\alpha}}_1\|_2\operatorname{sign}(\tilde{\boldsymbol{\alpha}}_1^{\mathrm{T}}\boldsymbol{e}_1),\quad \sigma_1=\frac{1}{\|\tilde{\boldsymbol{\alpha}}_1-s_1\boldsymbol{e}_1\|},$$

则矩阵

$$\tilde{\boldsymbol{Q}}_1=\boldsymbol{I}_{n-1}-2\boldsymbol{u}_1\boldsymbol{u}_1^{\mathrm{T}}\in\mathbf{R}^{(n-1)\times(n-1)}$$

是正交的.再令

$$\boldsymbol{Q}_1=\begin{bmatrix}1 & \\ & \tilde{\boldsymbol{Q}}_1\end{bmatrix}\in\mathbf{R}^{n\times n},$$

则 $\boldsymbol{Q}_1$ 仍是正交的矩阵,而且

$$\boldsymbol{Q}_1\boldsymbol{A}\boldsymbol{Q}_1^{\mathrm{T}}=\begin{bmatrix} a_{11} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ s_1 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & \cdots & a_{3n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}.$$

一般地,对 $k=2,3,\cdots,n-2$,令 $\tilde{\boldsymbol{\alpha}}_k=(a_{k+1,k}^{(k-1)},\cdots,a_{nk}^{(k-1)})^{\mathrm{T}}\in\mathbf{R}^{n-k}$,如果取

$$\boldsymbol{u}_k=\sigma_k(\tilde{\boldsymbol{\alpha}}_k-s_k\boldsymbol{e}_k)\in\mathbf{R}^{n-k},$$

其中

$$\boldsymbol{e}_k=(0,\cdots,1,0,\cdots,0)^{\mathrm{T}}\in\mathbf{R}^{n-k},$$

$$s_k=-\|\tilde{\boldsymbol{\alpha}}_k\|_2\operatorname{sign}(\tilde{\boldsymbol{\alpha}}_k^{\mathrm{T}}\quad\boldsymbol{\rho}_k),$$

$$\sigma_k=\frac{1}{\|\tilde{\boldsymbol{\alpha}}_k-s_k\boldsymbol{e}_k\|},$$

则矩阵

$$\tilde{\boldsymbol{Q}}_k=\boldsymbol{I}_{n-k}-2\boldsymbol{u}_k\boldsymbol{u}_k^{\mathrm{T}}$$

是 $n-k$ 阶的正交矩阵,而且

$$\boldsymbol{Q}_k=\begin{bmatrix}\boldsymbol{I}_k & \\ & \tilde{\boldsymbol{Q}}_k\end{bmatrix}$$

是 n 阶的正交矩阵.于是这一系列正交变换的结果是

$$H = Q_{n-2}\cdots Q_1 A Q_1^{\mathrm{T}}\cdots Q_{n-2}^{\mathrm{T}},$$

其中 H 是一个上海森伯格矩阵.当然它与 A 有完全一样的特征值.

4.4.3 海森伯格矩阵的 QR 分解

对(4-5)式的上海森伯格矩阵可以用(4-4)式中的平面旋转矩阵给出其 QR 分解.首先取

$$P_1 = P_{(1,2)} = \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix},$$

其中

$$c_1 = h_{11}/v_1, \quad s_2 = h_{21}/v_1, \quad v_1 = ((h_{11})^2 + (h_{21})^2)^{1/2}.$$

P_1 是正交的,它从左作用在 H 上实际仅改变它的前两行,即

$$P_1 H = \begin{bmatrix} v_1 & h_{12}^{(2)} & \cdots & \cdots & h_{1n}^{(2)} \\ 0 & h_{22}^{(2)} & \cdots & \cdots & h_{2n}^{(2)} \\ & h_{32} & \cdots & \cdots & h_{3n} \\ & & \ddots & \vdots & \vdots \\ & & & h_{n,n-1} & h_{nn} \end{bmatrix}.$$

一般地,对 $k = 2,3,\cdots,n-1$,取

$$P_k = P_{(k,k+1)} = \begin{bmatrix} I_{k-1} & & & \\ & c_k & s_k & \\ & -s_k & c_k & \\ & & & I_{n-k-1} \end{bmatrix},$$

其中

$$c_k = h_{kk}^{(k)}/v_k, \quad s_k = h_{k+1,k}/v_k, \quad v_k = [(h_{kk}^{(k)})^2 + (h_{k+1,k})^2]^{1/2}.$$

最终得到

$$P_{n-1}P_{n-2}\cdots P_2 P_1 H = R, \tag{4-6}$$

其中 R 是上三角矩阵.令 $Q = P_1^T P_2^T \cdots P_{n-1}^T$,$Q$ 是正交矩阵,则 $H = QR$,给出 H 的 QR 分解.

4.4.4 实用的 QR 算法

设 H 是一个上海森伯格矩阵,记 $H_1 = H$,则(4-6)式给出了它的 QR 分解.考虑 H 的 QR 算法,即

$$H_2 = RQ = RP_1^{\mathrm{T}}P_2^{\mathrm{T}}\cdots P_{n-1}^{\mathrm{T}}.$$

不难直接验证 H_2 还是一个上海森伯格矩阵.用数学归纳法就可以证明,H 的 QR

算法生成一个矩阵的序列$\{H_k\}$,其中每个H_k都是上海森伯格矩阵.

可见,计算矩阵A的特征值的QR算法包含如下两个步骤:

(1) 将A化成上海森伯格矩阵H.

(2) 对H作QR算法,生成一个上海森伯格矩阵的序列$\{H_k\}$.

对中小规模的矩阵,这是一种很有效的算法.

4.5　兰乔斯方法

本节讨论的问题是大型稀疏对称矩阵.由上一节的结果知,对任意的实方阵A,总存在正交矩阵Q和上海森伯格矩阵H,使

$$H = QAQ^{\mathrm{T}}.$$

由于A是对称的,所以H也是对称的,故H是三对角的.但如果A是稀疏的,将A化成三对角的过程中其稀疏性可能被破坏,从而导致算法计算过程中存储量过大.

4.5.1　兰乔斯过程

兰乔斯(Lanczos)过程是将对称矩阵正交相似地化为对称三对角矩阵的算法,其特点是在整个过程中原矩阵A保持不变.

设A是n阶的实对称矩阵,则存在n阶的正交矩阵Q,使得

$$Q^{\mathrm{T}}AQ = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}. \tag{4-7}$$

记$Q = (q_1, q_2, \cdots, q_n)$,兰乔斯过程的目标就是确定上式中的$\{\alpha_i, \beta_i\}$及正交矩阵$Q$.取一个$q \in \mathbf{R}^n$,且$\|q\|_2 = 1$,则

$$\alpha_1 = (Aq_1, q_1), \quad \beta_1 = \|Aq_1 - \alpha_1 q_1\|_2.$$

矩阵Q的第2列

$$q_2 = \frac{1}{\beta_1}(Aq_1 - \alpha_1 q_1).$$

一般地,对$k = 2, 3, \cdots, n-1$,有

$$\alpha_k = (Aq_k, q_k),$$
$$\beta_k = \|Aq_k - \alpha_k q_k - \beta_{k-1} q_{k-1}\|_2,$$
$$q_{k+1} = \frac{1}{\beta_k}(Aq_k - \alpha_k q_k - \beta_{k-1} q_{k-1}).$$

最后

$$\alpha_n = (Aq_n, q_n).$$

尽管A是稀疏的,但通常Q矩阵是满的.当后续计算中不需要Q的信息时,兰乔斯过程只需存最新的三个列$\{q_{k-1}, q_k, q_{k+1}\}$,所以这时兰乔斯过程存储的开销并不大.兰乔斯过程还可以写成矩阵形式:

$$AQ_k = Q_k T_k + r_k (e_k)^{\mathrm{T}}, \tag{4-8}$$

其中

$$
\begin{aligned}
e_k &= (0,\cdots,0,1)^{\mathrm{T}} \in \mathbf{R}^n, \quad r_k \in \mathbf{R}^n, \\
r_k &= Aq_k - \alpha_k q_k - \beta_{k-1} q_{k-1}, \\
Q_k &= (q_1, q_2, \cdots, q_k) \in \mathbf{R}^{n\times k}, \\
T_k &= \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}.
\end{aligned}
$$

4.5.2 兰乔斯算法

设 A 是 n 阶对称矩阵，其特征值记为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n,$$

它们相应的特征向量为 $z_1, z_2, \cdots, z_n$，而且 $(z_i, z_j) = \delta_{ij}, i, j = 1, 2, \cdots n$. T_m 是兰乔斯过程第 $m(m < n)$ 步生成的对称三对角矩阵，其特征值为

$$\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m.$$

$\{\theta_j\}$ 称为 A 的里兹值.严格的理论分析表明

$$\lambda_1 \geqslant \theta_1 \geqslant \lambda_1 - (\lambda_1 - \lambda_n)\frac{\tan^2\varphi_1}{(C_{m-1}(1+2\rho_1))^2},$$

$$\lambda_n \leqslant \theta_m \leqslant \lambda_n + (\lambda_1 - \lambda_n)\frac{\tan^2\varphi_n}{(C_{m-1}(1+2\rho_n))^2},$$

其中 $\cos\varphi_1 = |(q_1, z_1)|, \cos\varphi_n = |(q_1, z_n)|$; $C_{m-1}(x)$ 是 $m-1$ 次的切比雪夫多项式，且设

$$\rho_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n} > 0, \quad \rho_n = \frac{\lambda_{n-1} - \lambda_n}{\lambda_1 - \lambda_{n-1}} > 0.$$

当多项式的次数增加时，切比雪夫多项式在 $1 + 2\rho_1 > 1$ 及 $1 + 2\rho_n > 1$ 处的值是迅速增大的，所以 A 的极端里兹值能给出 A 的极端特征值很好的近似.由此得到切比雪夫算法基本步骤如下：

切比雪夫算法：设 A 是 n 阶对称的，

步 1　进行 m 步切比雪夫过程$(m < n)$，得矩阵 T_m.

步 2　计算 T_m 的特征值(即 A 的 Ritz 值)

$$\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m.$$

步 3　继续切比雪夫过程得 $T_s(m < s < n)$.

步 4　计算 T_s 的特征值

$$\tau_1 \geqslant \tau_2 \geqslant \cdots \geqslant \tau_s.$$

步 5　如果 $|\theta_1 - \tau_1|$ 及 $|\theta_m - \tau_s|$ 充分小，停止；否则令 $m := s$，返回步 3.

切比雪夫算法是计算对称矩阵极端特征值(即最大和最小的)的有效方法.矩阵 A 的次大里兹值和次小里兹值也可以用来近似 A 的次大和次小特征值，如 θ_2 近

似 λ_2，θ_3 近似 λ_3，θ_{m-1} 近似 λ_{n-1} 及 θ_{m-2} 近似 λ_{n-2}，但效果不及最极端的情况.

此外，算法在实际应用中，在此原理的基础上有许多变形，该原理也是计算特征值问题新算法的重要生长点.

4.6 豪斯霍尔德方法

本节仍假设矩阵 A 是对称的，其特征值为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n.$$

兰乔斯算法对极端特征值的计算是有效的，但对中间的部分特征值或计算全部特征值，则不一定很有效. 本节介绍的豪斯霍尔德(Householder)方法是特别针对中间特征值的.

由于 A 总可以正交相似于一个对称三对角矩阵，故不妨设

$$A = \begin{bmatrix} a_1 & b_2 & & \\ b_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_n \\ & & b_n & a_n \end{bmatrix},$$

而且可以设 $b_i \neq 0, i = 2,3,\cdots,n$，否则 A 可以分解为更低阶的三对角矩阵.

记 T_r 为 A 的 r 阶顺序主子矩阵，则

$$P_r(\lambda) = \det(T_r - \lambda I)$$

是关于 λ 的 r 次多项式. 如果记

$$P_0(\lambda) = 1, \quad P_{-1}(\lambda) = 0,$$

则有如下的递推关系：

$$P_r(\lambda) = (a_1 - \lambda)P_{r-1}(\lambda) - (b_r)^2 P_{r-2}(\lambda).$$

计算 A 的特征值就是要得到 $P_n(\lambda)$ 的根.

$\{P_k(\lambda)\}_{k=0}^{n}$ 是前面给出的多项式序列，μ 是一个实数. 定义

$$\operatorname{sgn}(P_i(\mu)) = \begin{cases} P_i(\mu)\text{ 的符号}, & \text{若 } P_i(\mu) \neq 0, \\ P_{i-1}(\mu)\text{ 的符号}, & \text{若 } P_i(\mu) = 0. \end{cases}$$

由于 $P_i(\lambda)$ 的根 $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_i$ 与 $P_{i-1}(\lambda)$ 的根 $\eta_1 \geqslant \eta_2 \geqslant \cdots \geqslant \eta_{i-1}$ 之间总满足

$$\mu_1 > \eta_1 > \mu_2 > \eta_2 > \cdots > \eta_{i-1} > \mu_i,$$

所以若有 μ 使 $P_i(\mu) = 0$，则 $P_{i-1}(\mu)$ 一定不为零，故 $\operatorname{sgn}(P_i(\mu))$ 总是有定义的. 如下结果构成了豪斯霍尔德算法的理论基础：有序集

$$E = \{+, \operatorname{sgn}(P_1(\mu)), \operatorname{sgn}(P_2(\mu)), \cdots, \operatorname{sgn}(P_s(\mu))\}$$

中相邻元素符号变化的次数等于多项式 $P_s(\lambda)$ 小于 μ 的根的个数.

注意：也可以考虑 E 中相邻元素符号相同的个数，它等于 $P_s(\lambda)$ 大于等于 μ 的根的个数.

豪斯霍尔德算法：

步 1　将 A 正交相似地变换为对称三对角矩阵.

步2 给定区间$[a,b]$，依据$\{P_i(a)\}_{i=0}^n$和$\{P_i(b)\}_{i=0}^n$的变号数，计算出$[a,b]$中特征值的个数.

步3 不断等分$[a,b]$中有特征值存在的子区间，直到区间的长度达到误差限.

4.7 广义特征值问题

广义特征值问题常见的形式为

$$Ax = \lambda Bx. \tag{4-9}$$

它是某些工程中常见的问题. 如果 A 是对称的，B 是对称正定的，这时求解方程(4-9)的通常方法是将它转化为一般的特征值问题.

由于 B 是正定的，故它存在楚列斯基分解

$$B = LL^{\mathrm{T}},$$

其中 L 是实的非奇异下三角矩阵. 于是方程(4-9)等价于

$$L^{-1}A(L^{-1})^{\mathrm{T}}L^{\mathrm{T}}x = \lambda L^{\mathrm{T}}x.$$

如果令

$$C = L^{-1}A(L^{-1})^{\mathrm{T}}, \quad y = L^{\mathrm{T}}x,$$

则方程(4-9)又等价于

$$Cy = \lambda y.$$

对小规模的问题，上面给出的是一种可行的算法，但若问题规模较大且带有特殊结构或稀疏性等，上述算法无疑会破坏这种结构. 进一步的讨论请参考文献[4,6]和[7].

参考文献

1 蔡大用. 数值代数. 北京：清华大学出版社，1987.
2 蔡大用，白峰杉. 高等数值分析. 北京：清华大学出版社，1997.
3 George A, Liu JW. Computer solution of large sparse positive definite systems. New Jersey: Prentice-Hall Inc, 1981.
4 Golub G H, Yon Lean C.F. Matrix computation. John Hopkins Univ Press, 1983.
5 Kelley C T. Iterative methods for linear and nonlinear equation. Philade lphia: SIAM, 1995.
6 Saod Y. Numerical methods for large eigenualue problems. Manchester: Manchester University Press, 1992.
7 Wilkin son J H. The algebraic eigenvalue problems. Clarendon Press, 1965.
8 徐树方. 矩阵计算的理论与方法. 北京：北京大学出版社，1995.

数学软件 Matlab 介绍

Matlab 是英文 Matrix Laboratory 的缩写，它是一个适用于多种类型计算机的软件平台，它的基本数据单元是不需指定维数的矩阵．可以将 Matlab 看成是一种计算机语言，但它使用起来要比其它高级语言简单且方便．Matlab 具有很强的编程和绘图功能，这里不拟对该软件作全面的介绍，而只介绍一下它在矩阵运算和线性方程组求解及特征值问题方面的功能，更深入的了解可以借助它的“help”功能或参考其它的资料．

1　矩阵的输入

生成小规模矩阵的基本方法是直接键盘输入．在命令窗口键入

A = [2　3　5; 3　4　2; 5　3　1]　　(F-1)

回车后会在命令窗口显示如下结果：

```
A =
    2  3  5
    3  4  2
    5  3  1
```

即变量 A 被赋值为一个三阶的方阵．如果(F-1)式的最后加上“;”，执行的结果仍是 A 被赋值为一个三阶方阵，但不显示在命令窗口上．

形成大规模矩阵的常用方法是用“load”语句从数据文件直接读入．若 fort.1 是一个纯数据文件，该文件共有 m 行，各行之间用回车分割，而同一行上数据两两之间用空格分开，且每一行数据的个数都为 n 个，则

load fort.1;

执行的结果是形成一个 $m \times n$ 的矩阵 fort.

2　特殊矩阵

Matlab 中提供几种常用的特殊矩阵：

ones(m, n)　生成 $m \times n$ 的矩阵，元素均为 1；

zeros(m, n)　生成 $m \times n$ 的零矩阵；

eye(n)　生成 n 阶的单位矩阵．

3　矩阵运算

假设 A 和 B 是两个 n 阶的方阵，则

$A + B$　给出 A、B 的和矩阵；

$A - B$　给出 A、B 的差矩阵；

$A * B$　给出 A、B 的乘积矩阵；

A' A 为实矩阵时，给出 A 的转置，

A 为复矩阵时，给出 A 的共轭转置.

4 线性方程组

首先介绍几个重要的函数，它们在方程组的求解中有重要价值.

hilb(n) 生成 n 阶的希尔伯特矩阵.这是一类著名的病态矩阵.

cond(A) 给出矩阵 A 条件数的估计.

rcond(A) 给出矩阵 A 条件数倒数的估计，与上述 cond(A) 相比，它速度更快但也更粗糙.

lu(A) 给出矩阵 A 的 LU 分解.

qr(A) 给出矩阵 A 的 QR 分解.

chol(A) 给出对称正定矩阵的 Cholesky 分解.

Matlab 中还提供了稀疏方程组的求解功能."sparse" 是起重要作用的函数，它有多种使用方法，其中之一是将矩阵存为稀疏形式.例如

$$A = \text{sparse}(\text{row},\text{col},\text{val},\text{n},\text{n});$$

其意义是：A 是一个 $n \times n$ 的矩阵，row，col 和 val 是维数相同的三个向量，A 在 $(i,j) = (\text{row}(k),\text{col}(k))$ 处的值为 val(k).注意，row，col，val 并不是稀疏矩阵的内部存储方式，而只是界面形式.

考虑线性代数方程组

$$Ax = b,$$

当然可以借助 lu、qr 等函数进行求解，但 Matlab 中还提供了更加简洁的方式：

$$x = A \backslash b;$$

或者

$$x = b/A.$$

它所采用的算法是根据矩阵 A 的性质由 Matlab 自适应选择的，方法有主元高斯消去法，QR 分解算法和楚列斯基算法等.如果 A 是存为稀疏形式的，使用的算法也是针对稀疏矩阵的.

5 矩阵特征值问题

Matlab 的基本函数为"eig"，它可以计算标准的特征值问题：

$$Au = \lambda u, \quad A \in \mathbf{R}^{n \times n},$$

也可以处理广义特征值问题：

$$Au = \lambda Bu, \quad A、B \in \mathbf{R}^{n \times n}.$$

针对不同的问题及不同的要求，函数"eig" 有如下四种调用方法：

l = eig(A); l 是一 n 元向量，其元素是 A 的特征值.

[u,l] = eig(A); u 是 n 阶矩阵，其列向量是 A 的特征向量；l 是对角矩阵，对角元素是其特征值.

l = eig(A,B); l 是 n 元向量，其元素是 (A,B) 的广义特征值.

[u,l] = eig(A,B); u 是 n 阶矩阵，其列向量是 (A,B) 的广义特征向量；l 是对角阵，其对角元素是其广义特征值.

6　其它软件和软件包

除 Matlab 外其它流行的软件平台还有 Mathematica 和 Maple 等，但数值计算方面的功能还是以 Matlab 最好，另外两个均更长于符号计算.

数值代数方面最早的两个软件包是 Linpack 和 Eispack，它们都是由矩阵运算基本程序库 BLAS 支持的.上述这几个软件包都是 FORTRAN 语言程序包. Linpack 的基本功能是线性代数方程组的数值求解，它所提供的基本方法是直接法；而 Eispack 是求解矩阵特征值问题的软件包. LApack 是在 Linpack 和 Eispack 的基础上发展起来的，它的功能更强大，也吸收了更多近代的算法，它有 FORTRAN 的版本，也有 C 的版本，使用上也更加灵活、方便.

·计算机数学卷·

第3篇

有限元法与边界元法

编　者　余德浩
审校者　崔俊芝

目　录

引　言

许多科学和工程问题可以通过不同的途径归结为不同的数学形式.它们或是表现为偏微分方程的边值问题,或是表现为区域上的变分问题,或是归结为边界上的积分方程或其它数学形式.这些问题的准确解通常是难以求得的.必须应用计算机近似地求解这些问题.这些不同的数学形式尽管在理论上是等价的,但在计算实践中却未必等效,它们往往导致不同的数值计算方法.

有限元方法是从等价的区域上的变分形式出发,近似地求解微分方程边值问题的一类非常重要的数值计算方法.有限元离散化的原始思想可追溯到20世纪40年代,从50年代起结构工程师便开始应用这一思想于实际计算.有限元数学理论的研究则直至60年代才开始.我国数学家冯康院士(1920—1993)在其研究小组的大量工程计算实践的基础上,于60年代初独立于西方系统地提出了有限元方法并奠定了其数学理论基础,为有限元法的创始和发展作出了历史性的贡献.

有限元法的创立是计算数学发展史上的一个里程碑.目前已在椭圆型偏微分方程数值求解方面占有主导地位.有限元法有广泛的适用性,特别适合几何或物理条件比较复杂的问题,便于程序标准化和工程应用.但有限元法也有局限性,例如难以处理无界区域问题及断裂或凹角区域上的问题.

与有限元法在区域上求解不同,边界元法则把微分方程边值问题归化为边界上的积分方程,然后利用各种离散化技术来求解.对微分方程作边界归化的思想可追溯到19世纪,但将它应用于数值计算却是从20世纪60年代开始的.有限元离散技术与边界归化理论相结合为这一方法的发展和应用打开了新局面,从而在70年代末被定名为边界元方法.

边界元法将问题降维处理,特别适于求解无界区域问题.

今天,有限元边界元法的内容已非常丰富,在经典方法的基础上,已发展出一系列新型方法,如非协调、非常规有限元法、自适应有限元边界元法、无限元方法、自然边界元方法、有限元边界元耦合算法、各种类型的区域分解算法,等等.各类有限元边界元法已被广泛地应用于结构工程、机械设计、土木建筑、水利建设、航空航天、石油勘探、大气海洋、核能技术等领域.

1　变分原理与剖分插值

有限元法的基础是变分原理与剖分插值.一方面,有限元法以一种大范围、全过程的数学分析即变分原理为出发点,它是传统的能量法,即里兹-伽辽金法的变形.另一方面,有限元法又采用分片多项式逼近来实现离散化过程,得到的代数方程组的系数矩阵是稀疏的,从而它又可看作差分方法的变形.有限元法正是这两类

方法相结合、取长补短而进一步发展的结果.

1.1 变分原理

椭圆型方程边值问题常有适当的变分原理与之等价.考察平面区域 Ω 上的二阶变系数椭圆型方程的混合边值问题:

$$\begin{cases} -\left(\frac{\partial}{\partial x}\beta\frac{\partial u}{\partial x}+\frac{\partial}{\partial y}\beta\frac{\partial u}{\partial y}\right)=f, & \Omega\text{ 内}, \\ u=u_0, & \Gamma_0\text{ 上}, \\ \beta\frac{\partial u}{\partial n}+\eta u=g, & \Gamma_1\text{ 上}, \end{cases} \tag{1-1}$$

其中 $\beta>0$ 及 f 为 Ω 上已知函数,u_0、η 及 g 为 Γ_0 或 Γ_1 上的已知函数,n 为边界 Γ 的外法线方向,在边界 Γ_0 上给定第一类边值,而在边界 Γ_1 上给定第三类边值.该边值问题等价于如下变分问题:

$$\begin{cases} J(u)=\inf\limits_{v\in H^1(\Omega)} J(v), \\ u=u_0, & \Gamma_0\text{ 上}, \end{cases} \tag{1-2}$$

其中

$$J(v)=\iint_\Omega\left(\frac{\beta}{2}\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2\right]-fu\right)\mathrm{d}x\mathrm{d}y+\int_{\Gamma_1}\left(\frac{1}{2}\eta u^2-gu\right)\mathrm{d}s \tag{1-3}$$

为能量积分,它是 u 及其一阶偏导数的二次泛函.

更一般些,设 $Q(u,v)$ 为如下对称双线性泛函:

$$Q(u,v)=\iint_\Omega[au_xv_x+b(u_xv_y+u_yv_x)+cu_yv_y+duv]\mathrm{d}x\mathrm{d}y+\int_{\Gamma_1}\alpha uv\mathrm{d}s, \tag{1-4}$$

其中 u_x 等表示偏导数,a、b、c、d 为 Ω 上已知函数,α 为 Γ_1 上的已知函数.如果 $a>0$, $ac>b^2$, $d\geqslant 0$, $\alpha\geqslant 0$, 则 $Q(u,v)$ 为半正定椭圆型双线性泛函.又若 $d>0$ 或 $\alpha>0$, 则 $Q(u,v)$ 正定.又设线性泛函

$$F(v)=\iint_\Omega fv\mathrm{d}x\mathrm{d}y+\int_{\Gamma_1}gv\mathrm{d}s, \tag{1-5}$$

其中 f 及 g 分别为 Ω 内及 Γ_1 上给定的函数,定义泛函

$$J(v)=\frac{1}{2}Q(v,v)-F(v), \tag{1-6}$$

于是变分问题

$$\begin{cases} J(u)=\inf\limits_{v\in H^1(\Omega)} J(v), \\ u=u_0, & \Gamma_0\text{ 上} \end{cases} \tag{1-7}$$

等价于虚功方程

$$\begin{cases} Q(u,v) = F(v), & \forall v \in H_0^1(\Omega), \\ u = u_0, & \Gamma_0 \text{上}, \end{cases} \tag{1-8}$$

其中 $H_0^1(\Omega) = \{v \in H^1(\Omega) \mid v = 0, \Gamma_0 \text{上}\}$.

这两种变分形式又都等价于微分方程边值问题

$$\begin{cases} -\mathrm{L}u = f, & \Omega \text{内}, \\ lu = g, & \Gamma_1 \text{上}, \\ u = u_0, & \Gamma_0 \text{上}, \end{cases} \tag{1-9}$$

其中

$$\mathrm{L}u = (au_x + bu_y)_x + (bu_x + cu_y)_y - \mathrm{d}u,$$

$$lu = \alpha u + [a\cos(n,x) + b\cos(n,y)]u_x + [b\cos(n,x) + c\cos(n,y)]u_y.$$

由能量泛函的极小化问题出发导致里兹(Ritz)法，而由虚功方程出发则导致伽辽金(Galerkin)法.后者比前者的适用范围更广泛.此二法的关键均为写出等价的变分形式及找到解函数空间的一组坐标函数系.

1.2 剖分插值

与经典的里兹-伽辽金法采用支集为全区域的坐标函数不同，有限元法将求解区域剖分为有限个单元，然后用分片插值函数来逼近解函数.剖分通常应满足一定的正规性条件，单元应具有简单的形状.在二维情况下常采用三角形或四边形.插值则分为拉格朗日型及埃尔米特型.近似函数空间的基函数都只有小支集.

在正规剖分的条件下，点元数、线元数与面元数之比对三角形剖分为

$$N_0 : N_1 : N_2 \approx 1:3:2, \tag{1-10}$$

而对四边形剖分则为

$$N_0 : N_1 : N_2 \approx 1:2:1. \tag{1-11}$$

三角形剖分下的有限元法常采用面积坐标$(\lambda_1,\lambda_2,\lambda_3)$，$\lambda_i \geqslant 0, i = 1,2,3$.面积坐标是一种局部坐标.若三角形三顶点的直角坐标分别为(x_1,y_1)，(x_2,y_2)及(x_3,y_3)，则有

$$\begin{cases} x = x_1\lambda_1 + x_2\lambda_2 + x_3\lambda_3, \\ y = y_1\lambda_1 + y_2\lambda_2 + y_3\lambda_3, \end{cases} \tag{1-12}$$

及

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{1-13}$$

1.2.1 三角形线性单元

以三角形三顶点为插值结点，在面积坐标下，基函数有最简单的表达式：

$$\begin{cases} L_1 = \lambda_1, \\ L_2 = \lambda_2, \\ L_3 = \lambda_3. \end{cases} \tag{1-14}$$

此时函数 $f(x,y)$ 的线性插值为

$$I_1 f(x,y) = \sum_{i=1}^{3} f(x_i, y_i)\lambda_i. \tag{1-15}$$

分片三角形线性插值有唯一可解性、整体连续性和如下逼近度：

$$| I_1 f - f |_{s,\Omega} \leqslant M h^{2-s} | f |_{2,\Omega}, \quad s = 0,1, \tag{1-16}$$

其中 M 为与 h 及 f 无关的常数，h 为该剖分下单元边长之最大值. 其总体自由度为点元数 N_0.

1.2.2 三角形二次单元

以三角形三顶点及三边中点为插值结点，在面积坐标下，基函数的表达式为

$$\begin{cases} L_1 = \lambda_1(2\lambda_1 - 1), \quad L_2 = \lambda_2(2\lambda_2 - 1), \\ L_3 = \lambda_3(2\lambda_3 - 1), \\ M_1 = 4\lambda_2\lambda_3, \quad M_2 = 4\lambda_1\lambda_3, \\ M_3 = 4\lambda_1\lambda_2. \end{cases} \tag{1-17}$$

此时函数 $f(x,y)$ 的二次插值为

$$I_2 f(x,y) = \sum_{i=1}^{3} [f(A_i)L_i + f(B_i)M_i], \tag{1-18}$$

其中 A_i、B_i ($i = 1,2,3$) 分别为三角形的三顶点及三边中点. 分片三角形二次插值有唯一可解性、整体连续性和如下逼近度：

$$| I_2 f - f |_{s,\Omega} \leqslant M h^{3-s} | f |_{3,\Omega}, \quad s = 0,1. \tag{1-19}$$

其总体自由度为 $N_0 + N_1 \approx 4N_0$.

1.2.3 双线性矩形单元

以矩形四顶点为插值结点，在单元局部坐标 (ξ,η) 下，基函数的表达式为

$$\begin{cases} L_1 = \dfrac{1}{4}(1 - \xi)(1 - \eta), \\ L_2 = \dfrac{1}{4}(1 + \xi)(1 - \eta), \\ L_3 = \dfrac{1}{4}(1 + \xi)(1 + \eta), \\ L_4 = \dfrac{1}{4}(1 - \xi)(1 + \eta). \end{cases} \tag{1-20}$$

此时函数 $f(x,y)$ 的双线性插值为

$$\Pi_1 f(x,y) = \sum_{i=1}^{4} f(A_i)L_i, \tag{1-21}$$

其中 A_i ($i = 1,2,3,4$) 为矩形的四顶点. 分片矩形双线性插值有唯一可解性、整体连续性和如下逼近度：

$$| \Pi_1 f - f |_{s,\Omega} \leqslant M h^{2-s} | f |_{2,\Omega}, \quad s = 0,1, \tag{1-22}$$

其中 h 为单元的长边长. 其总体自由度为 N_0.

1.2.4 双二次矩形单元

以矩形四顶点、四边中点及形心为插值结点,在局部坐标下,基函数的表达式为

$$\begin{cases} L_1 = \frac{1}{4}\xi\eta(1-\xi)(1-\eta), \quad L_2 = \frac{1}{4}\xi\eta(1+\xi)(1-\eta), \\ L_3 = \frac{1}{4}\xi\eta(1+\xi)(1+\eta), \quad L_4 = \frac{1}{4}\xi\eta(1-\xi)(1+\eta), \\ L_C = (1-\xi^2)(1-\eta^2), \\ M_1 = \frac{1}{2}\xi(\xi-1)(1-\eta^2), \quad M_2 = \frac{1}{2}\eta(\eta-1)(1-\xi^2), \\ M_3 = \frac{1}{2}\xi(\xi+1)(1-\eta^2), \quad M_4 = \frac{1}{2}\eta(\eta+1)(1-\xi^2). \end{cases} \tag{1-23}$$

此时函数 $f(x,y)$ 的双二次插值为

$$\Pi_2 f(x,y) = \sum_{i=1}^{4}[f(A_i)L_i + f(B_i)M_i] + f(C)L_C, \tag{1-24}$$

其中 A_i、$B_i(i=1,2,3,4)$ 分别为矩形四顶点及四边中点,C 为形心.分片矩形双二次插值有唯一可解性、整体连续性和如下逼近度:

$$|\Pi_2 f - f|_{s,\Omega} \leqslant Mh^{3-s}|f|_{3,\Omega}, \quad s = 0,1. \tag{1-25}$$

其总体自由度为 $N_0 + N_1 + N_2 \approx 4N_0$.

以上均为拉格朗日型插值单元.

1.3 有限元离散化

考虑有界区域 Ω 上的变分问题(1-7)或(1-8),它们等价于微分方程边值问题(1-9).为简单起见,设 $u_0 = 0$.有限元离散化按如下步骤进行:

步 1 单元剖分.对求解区域作正规剖分,例如三角形单元剖分.区域边界被截弯取直.对点元编号,位于约束边界 Γ_0 上的点元因其值已知可不编在内.

步 2 选择插值方式.例如取分片线性插值.设 $\Lambda_i(x,y)$ 为对应于各点元 P_i,$i=1,2,\cdots,m$ 的满足齐次约束条件的插值基函数,即满足

$$\Lambda_i(P_j) = \delta_{ij} = \begin{cases}1, & i=j, \\ 0, & i\neq j,\end{cases} \tag{1-26}$$

及 $$\Lambda_i(P) = 0, \qquad \forall P \in \Gamma_0.$$

基函数全体 $\{\Lambda_i\}_{i=1}^{m}$ 线性组合成索伯列夫(Sobolev)空间 $H^1(\Omega)$ 的子空间 S_h.于是 $\bar{\Omega}$ 上任意满足齐次约束边界条件的分片线性函数均可表示为

$$V(x,y) = \sum_{i=1}^{m} V_i\Lambda_i(x,y). \tag{1-27}$$

$V(x,y)$ 在结点 P_i 取值 V_i.

步 3 建立离散方程.离散变分问题为

$$\begin{cases} \text{求 } u_h \in S_h\text{，使得} \\ J(u_h) = \min\limits_{v_h \in S_h} J(v_h), \end{cases} \tag{1-28}$$

或
$$\begin{cases} \text{求 } u_h \in S_h\text{，使得} \\ Q(u_h, v_h) = F(v_h), \forall\, v_h \in S_h. \end{cases} \tag{1-29}$$

二者均导致求解线性代数方程组

$$\sum_{j=1}^{m} Q(\Lambda_i, \Lambda_j) U_j = F(\Lambda_i), \quad i = 1, 2, \cdots, m. \tag{1-30}$$

其 $u_h = \sum\limits_{j=1}^{m} U_j \Lambda_j$ 即为有限元近似解.

令 $q_{ij} = Q(\Lambda_i, \Lambda_j)$，方程组(1-30)的系数矩阵 $Q = [q_{ij}]_{m\times m}$ 称为离散变分问题的**刚度矩阵**. 对上述由自共轭边值问题导出的离散变分问题，刚度矩阵有对称正定性：

$$q_{ij} = q_{ji},$$
$$\sum_{i,j} q_{ij} V_i V_j \geqslant 0,$$

且仅当 $V_j = 0 (j = 1, 2, \cdots, m)$ 时等式成立. 又由于基函数 Λ_j 的值仅在少数几个单元上非零，则刚度矩阵 Q 有高度稀疏性.

2 协调元的理论分析

若有限元解函数空间为原问题解空间的子空间，则称为**协调有限元**，否则称为**非协调有限元**.

有限元法的理论分析通常在索伯列夫空间的框架下进行.

2.1 索伯列夫空间初步

设 Ω 是 N 维欧氏空间 $\mathbf{R}^N$ 中的开集，$x \in \mathbf{R}^N$，$x = (x_1, x_2, \cdots, x_N)$. $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_N\}$ 为多重指标，α_i 为非负整数. 记

$$D^\alpha = \frac{\partial^{\alpha_1 + \cdots + \alpha_N}}{\partial x_1^{\alpha_1} \cdots \partial x_N^{\alpha_N}}, \quad |\alpha| = \alpha_1 + \cdots + \alpha_N.$$

对定义在 Ω 上的函数 $f(x)$，其支集 $\mathrm{supp} f(x)$ 是使得 $f(x) \neq 0$ 的 x 点集的闭包. 记 Ω 内 m 次连续可微函数的全体为 $C^m(\Omega)$，其中 m 是非负整数或 ∞. $C^m(\Omega)$ 中所有在 Ω 内有紧支集的函数的集合记作 $C_0^m(\Omega)$.

如果 $u \in C^k(\bar{\Omega})$，$v \in C_0^\infty(\Omega)$，多次应用格林(Green)公式可得

$$\int_\Omega v D^\alpha u \mathrm{d}x = (-1)^{|\alpha|} \int_\Omega u D^\alpha v \mathrm{d}x, \quad \forall\, v \in C_0^\infty(\Omega), |\alpha| \leqslant k.$$

利用此式可推广导数的概念.

定义1 设函数 $u \in L_2(\Omega)$，如果存在 $u^{(\alpha)} \in L_2(\Omega)$，使得对一切 $v \in C_0^\infty(\Omega)$ 有

$$\int_\Omega v u^{(\alpha)} \mathrm{d}x = (-1)^{|\alpha|} \int_\Omega u \mathrm{D}^\alpha v \mathrm{d}x, \quad \forall v \in C_0^\infty(\Omega), \tag{2-1}$$

则称 $u^{(\alpha)}$ 是 u 的 $|\alpha|$ 阶广义导数，仍记作 $\mathrm{D}^\alpha u$.

广义导数在几乎处处相等的意义下唯一. 经典连续导数必是广义导数，但反之不然.

在函数空间 $C^k(\bar{\Omega})$ 上定义内积:

$$(u, v)_k = \int_\Omega \sum_{|\alpha|=0}^{k} \mathrm{D}^\alpha u \mathrm{D}^\alpha v \mathrm{d}x, \tag{2-2}$$

其中求和符号表示对一切 $\alpha = \{\alpha_1, \cdots, \alpha_N\}$，$0 \leqslant |\alpha| \leqslant k$，求和. u 的范数（或称为模）定义为

$$\| u \|_{k,\Omega} = \sqrt{(u, u)_k}. \tag{2-3}$$

$C^k(\bar{\Omega})$ 在此内积下成为内积空间，但并不完备，即对于任意 $C^k(\bar{\Omega})$ 中的基本列 $\{u_n\}$，未必能在 $C^k(\bar{\Omega})$ 中找到极限元 u，使得 $\| u_n - u \|_{k,\Omega} \to 0 (n \to \infty)$.

记 $C^k(\bar{\Omega})$ 在范数 $\| \cdot \|_k$ 意义下完备化得到的函数空间为 $H^k(\Omega)$，$C_0^k(\Omega)$ 在范数 $\| \cdot \|_k$ 意义下得到的完备化空间为 $H_0^k(\Omega)$，显然 $H_0^k(\Omega)$ 是 $H^k(\Omega)$ 的子空间. 索伯列夫空间 $H^k(\Omega)$ 及 $H_0^k(\Omega)$ 均为完备的内积空间，即希尔伯特空间.

$W^{k,p}(\Omega)$ 表示本身及其直到 k 阶的广义导数均属于 $L_p(\Omega)$ 的函数全体. 当 $p = 2$ 时，$W^{k,2}(\Omega) = H^k(\Omega)$.

定理1（嵌入定理1） 若 $k > l$，则 $H^k(\Omega)$ 紧嵌入于 $H^l(\Omega)$，记为

$$H^k(\Omega) \overset{C}{\hookrightarrow} H^l(\Omega). \tag{2-4}$$

定理2（嵌入定理2） 若有非负整数 l 满足 $k - l > \frac{n}{2}$，则 $H^k(\Omega)$ 紧嵌入于 $C^l(\bar{\Omega})$，记为

$$H^k(\Omega) \overset{C}{\hookrightarrow} C^l(\bar{\Omega}). \tag{2-5}$$

对于分片充分光滑的函数，若它在全区域属于 $C^k(\bar{\Omega})$，则必属于 $H^{k+1}(\Omega)$.

对 $u \in H^k(\Omega)$，定义

$$| u |_k = \left[\int_\Omega \sum_{|\alpha|=k} (\mathrm{D}^\alpha u)^2 \mathrm{d}x \right]^{\frac{1}{2}} \tag{2-6}$$

为 u 的 k 阶半模.

定理3 （等价模定理）若 $H^k(\Omega)(k \geqslant 1)$ 上的有界线性泛函 $l_1, l_2, \cdots, l_M$ 对于次数不高于 $k-1$ 次的任何非零多项式不同时为0，则模 $\| u \|_k$ 与模 $| u |_k + \sum_{i=1}^{M} | l_i(u) |$ 等价，即存在正常数 α 与 β，使对一切 $u \in H^k(\Omega)$，成立

$$\alpha \| u \|_k \leqslant | u |_k + \sum_{i=1}^{M} | l_i(u) | \leqslant \beta \| u \|_k. \tag{2-7}$$

由此定理可得如下两个很有用的不等式.

庞加莱(Poincaré)不等式:若 $u \in H^1(\Omega)$,则存在常数 c,使得

$$\| u \|_{1,\Omega} \leqslant c(| u |_{1,\Omega} + | \int_{\Omega} u \mathrm{d}x |). \tag{2-8}$$

弗里德里希斯(Friedrichs)不等式:若 $u \in H^1(\Omega)$,则存在常数 c,使得

$$\| u \|_{1,\Omega} \leqslant c(| u |_{1,\Omega} + | \int_{\partial\Omega} u \mathrm{d}s |). \tag{2-9}$$

特别地,当 $u \in H_0^1(\Omega)$ 时,

$$\| u \|_{1,\Omega} \leqslant c | u |_{1,\Omega}.$$

2.2　变分问题的适定性

微分方程边值问题或变分问题的解的存在性、唯一性和稳定性(也即解对定解条件的连续依赖性)统称为**适定性**.在用有限元法求解微分方程边值问题前,应首先研究相应的变分问题及其离散问题的适定性.

定理4　(拉克斯 - 米尔格兰姆((Lax-Milgram)定理)设 H 是希尔伯特空间,$B(u,v)$ 是 H 上的双线性泛函,且满足

$$| B(u,v) | \leqslant M \| u \| \, \| v \| \text{(连续性,有界性)},$$

$$B(u,v) \geqslant \gamma \| v \|^2 \qquad \text{(强制性,V- 椭圆性)},$$

其中 M、γ 为正常数,又 $F(v)$ 是 H 上有界线性泛函,则存在唯一的 $u \in H$,使得

$$B(u,v) = F(v), \qquad \forall v \in H,$$

且有估计

$$\| u \| \leqslant \frac{1}{\gamma} \| F \|.$$

应用拉克斯 - 米尔格兰姆定理可以证明如下问题的适定性.

例1　泊松方程第一边值问题:

$$\begin{cases} -\Delta u = f, & \Omega \text{ 内}, \\ u = u_0, & \partial\Omega \text{ 上}. \end{cases}$$

取希尔伯特空间 $H = H_0^1(\Omega)$,令 $w = u - u_0$,则 $w \in H_0^1(\Omega)$.按(1-4)式的定义,上述问题等价于

$$\begin{cases} \text{求 } w \in H_0^1(\Omega), \text{使得} \\ Q(w,v) = (f,v) - Q(u_0,v), \quad \forall v \in H_0^1(\Omega), \end{cases}$$

再应用拉克斯 - 米尔格兰姆定理,即证.

例2　第三边值问题:

$$\begin{cases} -\Delta u + qu = f, & \Omega \text{ 内}, \\ \dfrac{\partial u}{\partial n} + bu = g, & \partial\Omega \text{ 上}, \end{cases}$$

其中 Ω 为有界区域,q、b 有界,$q \geqslant q_0 > 0, b \geqslant 0$,或 $q \geqslant 0, b \geqslant b_0 > 0$.

取 $H = H^1(\Omega)$,应用拉克斯 - 米尔格兰姆定理即证.

例3　若第二边值问题

$$\begin{cases} -\Delta u = f, & \Omega \text{ 内}, \\ \dfrac{\partial u}{\partial n} = g, & \partial\Omega \text{ 上} \end{cases}$$

满足相容性条件 $\iint\limits_{\Omega} f\mathrm{d}x\mathrm{d}y + \int\limits_{\partial\Omega} g\mathrm{d}s = 0$，则解存在且可差一任意常数.

在子空间 $H^* = \{v \in H^1(\Omega) \mid \iint\limits_{\Omega} v\mathrm{d}x\mathrm{d}y = 0\}$ 中考虑变分问题并应用拉克斯-米尔格兰姆定理，即证.

2.3　收敛性与误差估计

设 u 及 u_h 分别为原问题的准确解及有限元法近似解.

定理 5　（投影定理）当有限元解空间 S_h 是原问题解空间 H 的子空间时，则

$$Q(u - u_h, v_h) = 0, \qquad \forall v_h \in S_h, \tag{2-10}$$

$$\| u - u_h \|_Q = \inf_{v_h \in S_h} \| u - v_h \|_Q. \tag{2-11}$$

推论　$Q(u_h, u_h) \leqslant Q(u, u)$.

上式表明相应于有限元近似解的应变能一定不大于真实的应变能.

定理 6　（抽象误差估计）设 H 为希尔伯特空间，S_h 为其线性子空间，$u \in H$ 及 $u_h \in S_h$ 分别为如下问题的解：

$$Q(u, v) = F(v), \qquad \forall v \in H,$$
$$Q(u_h, v_h) = F(v_h), \qquad \forall v_h \in S_h,$$

其中双线性泛函 Q 与线性泛函 F 满足拉克斯-米尔格兰姆定理的条件，则存在与 S_h 无关的常数 c，使得

$$\| u - u_h \|_H \leqslant c \inf_{v_h \in S_h} \| u - v_h \|_H. \tag{2-12}$$

下面特别设 $u \in H_0^1(\Omega)$ 是泊松方程齐次第一边值问题的广义解：

$$Q(u, v) = (f, v), \qquad \forall v \in H_0^1(\Omega),$$

其中

$$Q(u, v) = \iint\limits_{\Omega} \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) \mathrm{d}x\mathrm{d}y,$$

而 $u_h \in S_h \subset H_0^1(\Omega)$ 是其有限元近似解：

$$Q(u_h, v_h) = (f, v_h), \qquad \forall v_h \in S_h.$$

如果区域剖分和插值选取满足插值逼近定理的条件，则有下列结果：

定理 7　（收敛性）

$$\lim_{h \to 0} \| u - u_h \|_Q = 0. \tag{2-13}$$

定理 8　（能量模估计）若 $u \in H_0^1(\Omega) \cap H^{k+1}(\Omega)$，且插值算子 $\Pi: H_0^1(\Omega) \cap H^{k+1}(\Omega) \to S_h$ 保持

$$\Pi p_k = p_k, \qquad \forall p_k \in P_k,$$

其中 P_k 为次数不高于 k 的多项式全体，则

$$\| u - u_h \|_Q \leqslant ch^k \| u \|_{k+1,\Omega}, \quad k \geqslant 1. \tag{2-14}$$

定理9 （L_2模估计）设 Ω 是凸多角形，若 $u \in H_0^1(\Omega) \cap H^{k+1}(\Omega)$，且算子Π满足定理8中条件，则

$$\| u - u_h \|_{L_2(\Omega)} \leqslant ch^{k+1} \| u \|_{k+1,\Omega}, \quad k \geqslant 1. \tag{2-15}$$

定理10 （L^∞ 模估计）对三角形剖分下的线性协调元，若 $u \in H_0^1(\Omega) \cap W^{2,\infty}(\Omega)$，则

$$\| u - u_h \|_{L^\infty(\Omega)} \leqslant ch^2 |\ln h|^{3/2} | u |_{2,\infty,\Omega}. \tag{2-16}$$

上述定理中 c 均为与 h 及 u 无关的常数.

3 其它类型的有限元法

除经典的协调有限元法外，有限元法还包括非协调有限元法、混合有限元法、杂交有限元法及无限元法.

3.1 非协调有限元法

若取有限元解函数空间不在原问题求解空间内，则称为非协调有限元. 非协调元法在实际问题中有很多应用，特别适用于求解四阶问题.

下面列举几个非协调元的例子.

Crouzeix-Raviart元. 这是一类三角形单元，节点参数为各边高斯点处的函数值，特别取1阶高斯点时，节点即为三角形三边中点.

Wilson元. 这是矩形单元，节点参数是函数在矩形单元四顶点处的值及两个二阶导数 $D_{11}v$、$D_{22}v$ 在单元上的平均值，在单元 K 上，$v_{h,K} \in P_2(K)$.

上述两例用于二阶问题，其有限元函数空间不属于 $C^0(\Omega)$. 下面是四阶问题的非协调单元.

Morley元. 这是一个三角形单元，节点参数是三角形三顶点处函数值及各边中点外法向导数值，$v_{h,K} \in P_2(K)$. 这不是 C^0 元.

De Veubeke元. 这是不完全三次插值三角形单元，节点参数是三角形三顶点及三边中点处的函数值以及三边上法向导数的平均值. 这是 C^0 元但不是 C^1 元.

Adini元. 这是一种不完全双三次插值矩形单元，节点参数取矩形四顶点函数值及两个方向导数值，共12个自由度. 由于在相邻单元的公共边上法向导数有跳跃，故也是非协调元.

在非协调元法中 $Q(u_h, v_h)$ 可能没有意义，故用

$$Q_h(u_h, v_h) = \sum_{i=1}^{N} Q_{k_i}(u_h, v_h) \tag{3-1}$$

代替 $Q(u_h, v_h)$，其中 N 为 Ω 中单元个数. 于是非协调元法就是

$$\begin{cases}\text{求 } u_h \in V_h, \text{使得} \\ Q_h(u_h, v_h) = F(v_h), \forall v_h \in V_h,\end{cases} \tag{3-2}$$

其中 V_h 为非协调元解函数空间.

定理 1 若 $\| v_h \|_{Q_h} = [Q_h(v_h, v_h)]^{\frac{1}{2}}$ 构成 V_h 空间中的范数,则由(3-2)式得到的代数方程组必有唯一解.

定理 2 若 u 为原问题的广义解,u_h 为相应的非协调元解,如果 u_h 按度量 $\| \cdot \|_{Q_h}$ 收敛到 u,那么必有

$$\Delta = \sup_{v_h \in V_h} \frac{| E_h(u, v_h) |}{\| v_h \|_{Q_h}} \longrightarrow 0 \quad (h \to 0), \tag{3-3}$$

其中

$$E_h(u, v_h) = Q_h(u, v_h) - F(v_h).$$

反之,若(3-3)式成立,且插值误差 $\| u - \Pi u \|_{Q_h} \to 0$,则 $\| u - u_h \|_{Q_h} \to 0$.

非协调元解的误差由两部分构成,第一部分是插值逼近误差,第二部分是由非协调性引起的误差.因为对协调元来说,$E_h(u, v_h) = 0$,非协调元解是否收敛将取决于当 $h \to 0$ 时 Δ 是否趋于 0.

定理 2 给出了非协调元收敛的充要条件,但这一条件并不容易验证,于是需要"广义分片检查"及更简便的"F-E-M-Test"(参见文献[6]).

3.2 混合有限元法

有限元法的基本思想是将微分方程的边值问题化为变分问题,然后离散化求解.但同一边值问题可以对应不同形式的变分问题.将微分方程降阶以后再化为变分问题便导致混合有限元法.

考虑边值问题

$$\begin{cases} -\Delta u = f, & \Omega \text{ 内}, \\ u = g, & \partial\Omega \text{ 上}. \end{cases} \tag{3-4}$$

引进新的未知函数 $p_i = \dfrac{\partial u}{\partial x_i}$,则方程化为

$$\left.\begin{cases} p_i = \dfrac{\partial u}{\partial x_i}, \quad i = 1, \cdots, n \\ -\displaystyle\sum_{i=1}^{n} \frac{\partial p_i}{\partial x_i} = f, \end{cases}\right. \quad (\Omega \text{ 内}). \tag{3-5}$$

由此便得如下形式的变分问题:

$$\begin{cases} \text{求 } \boldsymbol{p} \in (H^1(\Omega))^n, u \in L^2(\Omega), \text{使得} \\ a(\boldsymbol{p}, \boldsymbol{q}) + b(\boldsymbol{q}, u) = G(\boldsymbol{q}), & \forall \boldsymbol{q} \in (H^1(\Omega))^n, \\ b(\boldsymbol{p}, v) = F(v), & \forall v \in L^2(\Omega). \end{cases} \tag{3-6}$$

对于问题(3-6)可以引进泛函

$$J(q,v) = \frac{1}{2}a(q,q) + b(q,v) - G(q) - F(v), \tag{3-7}$$

于是问题(3-6)等价于泛函(3-7)的驻点问题.

上述变分问题的抽象提法是：设 X、Y 为希尔伯特空间，$a(p,q)$ 是 $X\times X$ 上的有界双线性泛函，$b(q,v)$ 是 $X\times Y$ 上的有界双线性泛函，$G(q)$ 是 X 上的有界线性泛函，$F(v)$ 是 Y 上的有界线性泛函，求 $p\in X$, $u\in Y$，使得

$$\begin{cases} a(p,q) + b(q,u) = G(q), & \forall q \in X, \\ b(p,v) = F(v), & \forall v \in Y. \end{cases} \tag{3-8}$$

其混合有限元近似为：取 X 的有限维子空间 X_h，Y 的有限维子空间 Y_h，求 $p_h\in X_h$, $u_h\in Y_h$，使得

$$\begin{cases} a(p_h,q_h) + b(q_h,u_h) = G(q_h), & \forall q_h \in X_h, \\ b(p_h,v_h) = F(v_h), & \forall v_h \in Y_h. \end{cases} \tag{3-9}$$

在 X_h 中取基函数 $\varphi_1,\varphi_2,\cdots,\varphi_N$，在 Y_h 中取基函数 $\psi_1,\psi_2,\cdots,\psi_M$. 令

$$p_h = \sum_{i=1}^{N} p_i\varphi_i,\quad u_h = \sum_{j=1}^{M} u_j\psi_j,$$

便可由(3-9)式得代数方程组：

$$\begin{cases} \boldsymbol{A}\boldsymbol{p} + \boldsymbol{B}\boldsymbol{u} = \boldsymbol{g}, \\ \boldsymbol{B}^{\mathrm{T}}\boldsymbol{p} = \boldsymbol{f}, \end{cases} \tag{3-10}$$

其中

$$\boldsymbol{p} = (p_1,p_2,\cdots,p_N)^{\mathrm{T}},\quad \boldsymbol{u} = (u_1,u_2,\cdots,u_M)^{\mathrm{T}},$$
$$\boldsymbol{g} = (g_1,g_2,\cdots,g_N)^{\mathrm{T}},\quad \boldsymbol{f} = (f_1,f_2,\cdots,f_M)^{\mathrm{T}}.$$

混合有限元法的优点是：它可以使方程降阶，适用于高阶方程；它在求出 $\boldsymbol{u}$ 的同时还求出 $\boldsymbol{p}$. 在力学中，$\boldsymbol{p}$ 往往表示应力，是重要的未知量.

混合有限元法的缺点是：方程复杂，代数方程组变得更庞大；两个空间 X_h 与 Y_h 的匹配很不容易，若选用不当会导致精度降低，甚至不收敛.

3.3　无限相似单元法与无限元法

经典有限元法将求解区域剖分为有限个有限大的单元，例如三角形或四边形，因此难以处理无界区域问题. 为剖分无界区域，必须用无限多个有限大的单元，或虽然只有有限个单元，但其中必有若干单元无限大. 于是在有限元法的基础上，于70年代又发展了无限相似单元法及无限元法.

设 Γ_0 是平面上的一个凸多边形，其外部区域为 Ω. 不妨设坐标原点在 Γ_0 内部. 为求解 Ω 上的边值问题，将 Ω 有规律地剖分为无限多个三角形单元：以原点为相似中心，取 $\xi>1$，以 $\xi,\xi^2,\cdots,\xi^k,\cdots$ 为比例常数，作 Γ_0 的相似形，分别记作 Γ_1, $\Gamma_2,\cdots,\Gamma_k,\cdots$；每两个多边形间的区域为一层，再将每一层进一步剖分成相似的若干个四边形，每个四边形又分为两个三角形，每层剖分方向一致. 于是得到无限相似剖分. 将各层刚度矩阵按节点叠加，可得一个无穷阶的总刚度矩阵，从而得无穷

阶的代数方程组.由于剖分的相似性,各层的刚度矩阵相同.利用这一特点,可应用特征系统方法或迭代法巧妙地求解上述无穷阶代数方程组(参见[3]).这便是无限相似单元法.

所谓无限元法则是指单元的几何形状是无限的,单元个数则是有限的.剖分的外层单元在某个方向没有外部边界,或者说边界在无限远.无限元法在单元划分上保持了灵活性,可以充分利用有限元法的解算技术和应用软件,其线性代数方程组的系数矩阵仍具有稀疏、对称、带状等特点.这一方法比无限相似单元法简单,便于推广应用.其缺点是关于无限单元的形函数需要特殊构造,要选择适当比较困难,且数值积分的计算量很大.

4 经典边界元法

边界元法是在经典的边界积分方程法的基础上吸取有限元离散化技术而发展起来的偏微分方程数值解法.它将所处理问题降低一维,只须在边界上进行单元剖分.这对无界区域问题特别有意义.

边界元法的基础是边界归化理论.边界归化的途径是多种多样的.从同一边值问题可得到许多不同的边界积分方程.这些积分方程可能是非奇异的,可能是弱奇异的,可能是柯西型奇异的,也可能是超奇异的,从而导致不同的边界元法.

经典边界归化方法通常分为间接边界归化与直接边界归化两种类型.

4.1 间接边界归化

从基本解及位势理论出发,许多椭圆型微分方程的边值问题可化为弗雷德霍姆(Fredholm)积分方程.此时积分方程的未知量不是原问题解的边值而是引入的新变量.

设 Ω 为以封闭曲线 Γ 为边界的二维有界区域.考察拉普拉斯(Laplace)方程的狄利克雷(Dirichlet)问题:

$$\begin{cases}\Delta u = 0, & \Omega \text{ 内}, \\ u = u_0, & \Gamma \text{ 上}, \end{cases} \tag{4-1}$$

及诺伊曼(Neumann)问题

$$\begin{cases}\Delta u = 0, & \Omega \text{ 内}, \\ \dfrac{\partial u}{\partial n} = g, & \Gamma \text{ 上}, \end{cases} \tag{4-2}$$

其中 n 为 Γ 上的外法线方向.边值问题(4-1)存在唯一解,而边值问题(4-2)在满足相容性条件:

$$\int_\Gamma g \mathrm{d}s = 0$$

时在差一个任意常数的意义下有唯一解.类似地,在 Ω 的补集的内部 Ω' 上也有拉普拉斯方程的狄利克雷问题及诺伊曼问题.对这两类外边值问题,必须对解在无穷

远处的性态作一定的限制才能保证解的唯一性，即要求解在无穷远处有界.

为建立解的积分表达式，要用到格林公式：

$$\int_{\Omega} v\Delta u \mathrm{d}x = \int_{\Gamma} v\frac{\partial u}{\partial n}\mathrm{d}s - \int_{\Omega}\nabla u \cdot \nabla v \mathrm{d}x, \tag{4-3}$$

及第二格林公式：

$$\int_{\Omega}(v\Delta u - u\Delta v)\mathrm{d}x = \int_{\Gamma}\left(v\frac{\partial u}{\partial n} - u\frac{\partial v}{\partial n}\right)\mathrm{d}s. \tag{4-4}$$

二维拉普拉斯方程的基本解为

$$E(x,y) = -\frac{1}{2\pi}\ln r, \tag{4-5}$$

其中 $r = |x-y| = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$，满足

$$-\Delta E(x,y) = \delta(x-y), \tag{4-6}$$

$\delta(\cdot)$ 为狄拉克(Dirac) 函数.

定理1　设 $\mathrm{int}\Gamma$ 和 $\mathrm{ext}\Gamma$ 分别表示 Γ 的内侧和外侧，法线方向 n 指向 Ω 外部，则拉普拉斯方程的解 u 有如下积分表达式：

$$u(y) = \frac{1}{2\pi}\int_{\Gamma}\left\{[u(x)]\frac{\partial}{\partial n_x}\ln|x-y| - \left[\frac{\partial u(x)}{\partial n}\right]\ln|x-y|\right\}\mathrm{d}s(x), \tag{4-7}$$

其中 $y \in \Omega \cup \Omega'$，

$$\left[\frac{\partial u}{\partial n}\right] = \left.\frac{\partial u}{\partial n}\right|_{\mathrm{int}\Gamma} - \left.\frac{\partial u}{\partial n}\right|_{\mathrm{ext}\Gamma},$$

$$[u] = u|_{\mathrm{int}\Gamma} - u|_{\mathrm{ext}\Gamma}.$$

引入辅助变量

$$\varphi = [u],\quad q = \left[\frac{\partial u}{\partial n}\right],$$

有如下结果：

二维拉普拉斯方程的解用单层位势表示：

$$u(y) = -\frac{1}{2\pi}\int_{\Gamma} q(x)\ln|x-y|\mathrm{d}s(x),\quad \forall y \in \mathbf{R}^2; \tag{4-8}$$

对狄利克雷问题，得到第一类弗雷德霍姆积分方程

$$-\frac{1}{2\pi}\int_{\Gamma} q(x)\ln|x-y|\mathrm{d}s(x) = u_0(y),\quad y \in \Gamma, \tag{4-9}$$

其积分核有 log 型弱奇异性；对诺伊曼问题，得到第二类弗雷德霍姆积分方程

$$\pm\frac{1}{2}q(y) - \frac{1}{2\pi}\int_{\Gamma} q(x)\frac{\partial}{\partial n_y}\ln|x-y|\mathrm{d}s(x) = g(y),\quad y \in \begin{matrix}\mathrm{int}\Gamma\\ \mathrm{ext}\Gamma\end{matrix}, \tag{4-10}$$

其积分核有柯西型奇异性.

二维拉普拉斯方程的解用双层位势表示：

$$u(y) = \frac{1}{2\pi}\int_{\Gamma}\varphi(x)\frac{\partial}{\partial n_x}\ln|x-y|\mathrm{d}s(x),\quad \forall y \in \Omega \cup \Omega', \tag{4-11}$$

对狄利克雷问题，得到第二类弗雷德霍姆积分方程

$$\pm\frac{1}{2}\varphi(y)+\frac{1}{2\pi}\int_{\Gamma}\varphi(x)\frac{\partial}{\partial n_x}\ln|x-y|\,\mathrm{d}s(x)=u_0(y),\quad y\in\begin{matrix}\mathrm{int}\Gamma\\ \mathrm{ext}\Gamma\end{matrix},\tag{4-12}$$

其积分核有柯西型奇异性;对诺伊曼问题,得到第一类积分方程

$$\frac{1}{2\pi}\int_{\Gamma}\varphi(x)\frac{\partial^2}{\partial n_y\partial n_x}\ln|x-y|\,\mathrm{d}s(x)=g(y),\quad y\in\Gamma,\tag{4-13}$$

其积分核有阿达马(Hadamard)型超奇异性.

对于三维区域 Ω 或 Ω' 内的拉普拉斯方程的第一类或第二类边值问题,只要应用三维拉普拉斯方程的基本解

$$E(x,y)=\frac{1}{4\pi|x-y|},\tag{4-14}$$

便可得如下结果.

三维拉普拉斯方程的解用单层位势表示:

$$u(y)=\frac{1}{4\pi}\int_{\Gamma}\frac{q(x)}{|x-y|}\mathrm{d}s(x),\quad\forall y\in\mathbf{R}^3,\tag{4-15}$$

对狄利克雷问题,得到第一类弗雷德霍姆积分方程

$$\frac{1}{4\pi}\int_{\Gamma}\frac{q(x)}{|x-y|}\mathrm{d}s(x)=u_0(y),\quad y\in\Gamma,\tag{4-16}$$

其积分核有弱奇异性;对诺伊曼问题,得到第二类弗雷德霍姆积分方程

$$\pm\frac{1}{2}q(y)+\frac{1}{4\pi}\int_{\Gamma}q(x)\frac{\partial}{\partial n_y}\left(\frac{1}{|x-y|}\right)\mathrm{d}s(x)=g(y),\quad y\in\begin{matrix}\mathrm{int}\Gamma\\ \mathrm{ext}\Gamma\end{matrix},\tag{4-17}$$

其积分核有柯西型奇异性.

三维拉普拉斯方程的解用双层位势表示:

$$u(y)=-\frac{1}{4\pi}\int_{\Gamma}\varphi(x)\frac{\partial}{\partial n_x}\left(\frac{1}{|x-y|}\right)\mathrm{d}s(x),\quad\forall y\in\Omega\cup\Omega',\tag{4-18}$$

对狄利克雷问题,得到第二类弗雷德霍姆积分方程

$$\pm\frac{1}{2}\varphi(y)-\frac{1}{4\pi}\int_{\Gamma}\varphi(x)\frac{\partial}{\partial n_x}\left(\frac{1}{|x-y|}\right)\mathrm{d}s(x)=u_0(y),\quad y\in\begin{matrix}\mathrm{int}\Gamma\\ \mathrm{ext}\Gamma\end{matrix},\tag{4-19}$$

其积分核有柯西型奇异性;对诺伊曼问题,得到第一类积分方程

$$-\frac{1}{4\pi}\int_{\Gamma}\varphi(x)\frac{\partial^2}{\partial n_y\partial n_x}\left(\frac{1}{|x-y|}\right)\mathrm{d}s(x)=g(y),\quad y\in\Gamma,\tag{4-20}$$

其积分核有阿达马型超奇异性.

由导出的边界积分方程求得 $q(x)$ 或 $\varphi(x)$ 后,再利用单层位势或双层位势表达式即可得原问题的解 $u(x)$.

4.2 直接边界归化

从基本解及格林公式出发,导致直接边界归化.直接法并不引入新的变量.积

分方程的未知量就是原问题未知量的边值.

仍以二维拉普拉斯方程的边值问题为例.在第二格林公式(4-4)内取 u 为所考察边值问题之解,v 为基本解(4-5),立即得到解的积分表达式

$$u(y)=\frac{1}{2\pi}\int_{\Gamma}\left\{u(x)\frac{\partial}{\partial n_x}\ln|x-y|-\frac{\partial u(x)}{\partial n}\ln|x-y|\right\}ds(x),\quad y\in\Omega,\tag{4-21}$$

以及在边界上

$$\frac{1}{2}u(y)=\frac{1}{2\pi}\int_{\Gamma}\left\{u(x)\frac{\partial}{\partial n_x}\ln|x-y|-\frac{\partial u(x)}{\partial n}\ln|x-y|\right\}ds(x),\quad y\in\Gamma.\tag{4-22}$$

对(4-21)式求法向导数还可得

$$\frac{\partial u(y)}{\partial n}=\frac{1}{2\pi}\int_{\Gamma}\left\{u(x)\frac{\partial^2}{\partial n_y\partial n_x}\ln|x-y|-\frac{\partial u(x)}{\partial n}\frac{\partial}{\partial n_y}\ln|x-y|\right\}ds(x),\quad y\in\Omega.\tag{4-23}$$

在(4-21)及(4-23)式中,令 y 趋向边界 Γ,注意到单层位势及双层位势越过边界时的连续性或跳跃性质,导致下列结果:

对拉普拉斯方程的狄利克雷问题,只须求解含log型弱奇异积分核的第一类弗雷德霍姆积分方程

$$\int_{\Gamma}u_n(x)E(x,y)ds(x)=\frac{1}{2}u_0(y)+\int_{\Gamma}u_0(x)\frac{\partial}{\partial n_x}E(x,y)ds(x),\quad y\in\Gamma,\tag{4-24}$$

或含柯西型奇异核的第二类弗雷德霍姆积分方程

$$\frac{1}{2}u_n(y)-\int_{\Gamma}u_n(x)\frac{\partial}{\partial n_y}E(x,y)ds(x)=-\int_{\Gamma}u_0(x)\frac{\partial^2}{\partial n_y\partial n_x}E(x,y)ds(x),y\in\Gamma,\tag{4-25}$$

对拉普拉斯方程的诺伊曼问题,只须求解含柯西型奇异核的第二类弗雷德霍姆积分方程

$$\frac{1}{2}u_0(y)+\int_{\Gamma}u_0(x)\frac{\partial}{\partial n_x}E(x,y)ds(x)=\int_{\Gamma}u_n(x)E(x,y)ds(x),\quad y\in\Gamma,\tag{4-26}$$

或含阿达马超奇异积分核的第一类积分方程

$$-\int_{\Gamma}u_0(x)\frac{\partial^2}{\partial n_y\partial n_x}E(x,y)ds(x)=\frac{1}{2}u_n(y)-\int_{\Gamma}u_n(x)\frac{\partial}{\partial n_y}E(x,y)ds(x),\quad y\in\Gamma.\tag{4-27}$$

上列诸式中右端已知,而左端含未知量 u_n 或 u_0.在解得 u_n 或 u_0 后,连同原已知边值 u_0 或 u_n 代入解的积分表达式

$$u(y)=\int_{\Gamma}\left\{u_n(x)E(x,y)-u_0(x)\frac{\partial}{\partial n_x}E(x,y)\right\}ds(x),\tag{4-28}$$

即得原问题的解 u.

注意，由于基本解并不唯一且实际上有无穷多个，例如拉普拉斯方程的基本解加上任意一个调和函数仍为基本解，故对同一边值问题通过间接或直接边界归化法可以得到无穷多个边界积分方程.

4.3 边界积分方程的数值求解

把边界剖分成单元.在二维情况通常取直线或圆弧段边界单元，在三维情况常取平面三角形或四边形边界单元.然后应用配置法或伽辽金法将边界积分方程离散化.

配置法以满足纯插值约束条件的方式寻求边界积分方程的近似解.例如求解边界积分方程

$$\int_{\Gamma} K(x,y)u_0(x)\mathrm{d}s(x) = f(y), \tag{4-29}$$

其中 $u_0(x)$ 为未知函数.设

$$u_{0h}(x) = \sum_{j=1}^{N} U_j\varphi_j(x), \tag{4-30}$$

其中 $\varphi_j(x), j = 1,\cdots,N$，为适当剖分下 Γ 上的插值基函数，$U_1,U_2,\cdots,U_N$ 为待定系数.将(4-30)式代入(4-29)式，并使其在适当的配置点 $y_1,\cdots,y_N$ 处成立，于是有如下线性代数方程组：

$$\sum_{j=1}^{N}\left[\int_{\Gamma} K(x,y_i)\varphi_j(x)\mathrm{d}s(x)\right]U_j = f(y_i), \quad i = 1,\cdots,N. \tag{4-31}$$

在求得 $U_j, j = 1,2,\cdots,N$，后即可由(4-30)式求得 $u_{0h}(x)$.它便是(4-29)式的配置法近似解.

配置法简单易行，计算量小，常被工程界应用，但不便于理论分析.

伽辽金法即有限元法在边界上的应用，是将边界积分方程化为边界上的变分问题然后离散化求解.由于有限元法已有成熟的理论，因此该法容易进行理论分析.缺点是需要计算许多二重积分，增大了计算量.

考察如下第一类弗雷德霍姆积分方程：

$$-\frac{1}{2\pi}\int_{\Gamma} q(x)\ln|x-y|\,\mathrm{d}s(x) = f(y), \quad y \in \Gamma. \tag{4-32}$$

令

$$Q(q,p) = -\frac{1}{2\pi}\int_{\Gamma}\int_{\Gamma}\ln|x-y|\,q(x)p(y)\mathrm{d}s(x)\mathrm{d}s(y),$$

于是(4-32)式等价于变分问题

$$\begin{cases} \text{求 } q(x) \in H^{-\frac{1}{2}}(\Gamma)\text{，使得} \\ Q(q,p) = \int_{\Gamma} fp\,\mathrm{d}s, \forall p \in H^{-\frac{1}{2}}(\Gamma), \end{cases} \tag{4-33}$$

其相应的离散变分问题为

$$\begin{cases} 求\ q_h(x) \in S_h, 使得 \\ Q(q_h, p_h) = \int_\Gamma f p_h \mathrm{d}s, \quad \forall p_h \in S_h, \end{cases} \tag{4-34}$$

其中 $S_h \subset H^{-\frac{1}{2}}(\Gamma)$，例如可取为 Γ 上分段常数函数空间或分段线性函数空间. 设 $L_i(s), i = 1, \cdots, N$，为 S_h 的基函数. 令

$$q_h = \sum_{j=1}^{N} q_j L_j(s),$$

便可由(4-34)式得到如下线性代数方程组：

$$\sum_{j=1}^{N} Q(L_j, L_i) q_j = \int_\Gamma f L_i \mathrm{d}s, \tag{4-35}$$

这里每一个系数 $Q(L_j, L_i)$ 是一个二重积分.

由于积分算子是非局部算子，无论应用配置法还是伽辽金法，得到的线性代数方程组的系数矩阵都是满矩阵.

5　自然边界元法

自然边界元法是我国学者首创的一种新型边界元法，它与国际流行的间接法或直接法完全不同，有很多独特的优点. 这一方法基于自然边界归化原理，将区域内的微分方程边值问题化为边界上的超奇异积分方程，然后在边界上离散化求解.

5.1　自然边界归化

自然边界归化是从格林公式及格林函数出发的一种特殊的直接边界归化方法. 其思想由冯康院士首先提出.

考察以光滑曲线 Γ 为边界的平面有界区域 Ω 中的拉普拉斯方程的诺伊曼边值问题：

$$\begin{cases} \Delta u = 0, \ \Omega 内, \\ \dfrac{\partial u}{\partial n} = u_n, \ \Gamma 上, \end{cases} \tag{5-1}$$

其中 $u_n \in H^{-\frac{1}{2}}(\Gamma)$，满足相容性条件

$$\int_\Gamma u_n \mathrm{d}s = 0.$$

边值问题(5-1)式等价于如下变分问题：

$$\begin{cases} 求\ u \in H^1(\Omega), 使得 \\ Q(u, v) = F(v), \forall v \in H^1(\Omega), \end{cases} \tag{5-2}$$

其中

$$Q(u, v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}x, \quad F(v) = \int_\Gamma u_n v \mathrm{d}s.$$

设 $G(x,y)$ 为格林函数,满足

$$\begin{cases} -\Delta G(x,y) = \delta(x-y), \\ G(x,y)\mid_{x\in\Gamma} = 0. \end{cases} \tag{5-3}$$

在格林第二公式

$$\int_\Omega (v\Delta u - u\Delta v)\mathrm{d}x = \int_\Gamma \left(v\frac{\partial u}{\partial n} - u\frac{\partial v}{\partial n}\right)\mathrm{d}s$$

中取 u 满足 $\Delta u = 0, v = G(x,y)$,即得拉普拉斯方程的解的积分表达式

$$u(y) = -\int_\Gamma u_0(x)\frac{\partial}{\partial n_x}G(x,y)\mathrm{d}s(x), \quad \forall y \in \Omega, \tag{5-4}$$

即泊松积分公式,其中 $u_0 = u\mid_\Gamma$.对(5-4)式取法向导数并令 y 由 Ω 内部趋向边界 Γ,便得自然积分方程

$$u_n(y) = -\int_\Gamma\left[\frac{\partial^2}{\partial n_y \partial n_x}G(x,y)\right]u_0(x)\mathrm{d}s(x), \quad y \in \Gamma. \tag{5-5}$$

算子 $\mathscr{K}: u_0 \to u_n$ 称**自然积分算子**,也称 **D-N 算子**.(5-5)中的积分核有超奇异性.设

$$\hat{Q}(u_0,v_0) = -\int_\Gamma\int_\Gamma \frac{\partial^2}{\partial n_y \partial n_x}G(x,y)u_0(x)v_0(y)\mathrm{d}s(x)\mathrm{d}s(y),$$

则自然积分方程(5-5)式等价于边界上的变分问题

$$\begin{cases} \text{求 } u_0 \in H^{\frac{1}{2}}(\Gamma), \text{使得} \\ \hat{Q}(u_0,v_0) = F(v_0), \forall v_0 \in H^{\frac{1}{2}}(\Gamma). \end{cases} \tag{5-6}$$

通过求解变分问题(5-6)式得到 u_0,再应用泊松积分公式(5-4)式,便得原问题的解 u.

自然积分算子是正数阶的拟微分算子,对单连通区域 Ω 上的调和边值问题,自然积分算子 $\mathscr{K}$ 满足

$$\mathscr{K}^2 = -\frac{\mathrm{d}^2}{\mathrm{d}s^2}, \tag{5-7}$$

其中 s 为 Γ 上的弧长参数.

5.2 典型区域的自然积分方程

某些椭圆型方程的解可以复变函数表示.

若 u 为 Ω 中调和方程(拉普拉斯方程)$\Delta u = 0$ 的解,则

$$u = \mathrm{Re}\varphi(z), \tag{5-8}$$

其中 $\varphi(z)$ 为 Ω 内解析函数,$z = x + \mathrm{i}y$.

若 u 为 Ω 中重调和方程 $\Delta^2 u = 0$ 的解,则

$$u = \mathrm{Re}[\varphi(z)\bar{z} + \psi(z)], \tag{5-9}$$

其中 $\varphi(z)$ 及 $\psi(z)$ 为 Ω 内解析函数,$\bar{z} = x - \mathrm{i}y$.

若 $\boldsymbol{u} = (u_1, u_2)$ 为 Ω 中平面弹性方程

$$\mu\Delta\boldsymbol{u} + (\lambda + \mu)\mathbf{grad}(\mathrm{div}\boldsymbol{u}) = 0$$

的解,则

$$\begin{cases} u_1 = \dfrac{1}{2\mu}\mathrm{Re}\left[\dfrac{\lambda + 3\mu}{\lambda + \mu}\varphi(z) - \bar{z}\varphi'(z) - \psi'(z)\right], \\ u_2 = \dfrac{1}{2\mu}\mathrm{Im}\left[\dfrac{\lambda + 3\mu}{\lambda + \mu}\varphi(z) + \bar{z}\varphi'(z) + \psi'(z)\right], \end{cases} \tag{5-10}$$

其中 $\varphi(z)$ 及 $\psi(z)$ 为 Ω 内解析函数.

若 $\boldsymbol{u} = (u_1, u_2)$ 及 p 为 Ω 中斯托克斯方程组

$$\begin{cases} -\nu\Delta\boldsymbol{u} + \mathbf{grad}p = 0, \\ \mathrm{div}\boldsymbol{u} = 0 \end{cases}$$

的解,则

$$\begin{cases} u_1 = \mathrm{Re}[-\varphi'(z)\bar{z} + \varphi(z) - \psi(z)], \\ u_2 = \mathrm{Im}[\varphi'(z)\bar{z} + \varphi(z) + \psi(z)], \\ p = -4\nu\mathrm{Re}\varphi'(z), \end{cases} \tag{5-11}$$

其中 $\varphi(z)$ 及 $\psi(z)$ 为 Ω 内解析函数.

应用上列复变函数表示、格林函数或傅里叶分析方法,可以得到若干典型区域上典型边值问题的自然积分方程与泊松积分公式的具体表达式.例如,

上半平面调和边值问题的泊松积分公式为

$$u(y_1, y_2) = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{y_2}{(y_1 - x_1)^2 + y_2^2}u_0(x_1)\mathrm{d}x_1, \quad y_2 > 0, \tag{5-12}$$

自然积分方程为

$$u_n(y_1) = -\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{u_0(x_1)}{(y_1 - x_1)^2}\mathrm{d}x_1. \tag{5-13}$$

半径为 R 的圆内区域调和边值问题的泊松积分公式为

$$u(r,\theta) = \frac{R^2 - r^2}{2\pi}\int_0^{2\pi}\frac{u_0(\theta')}{R^2 + r^2 - 2Rr\cos(\theta - \theta')}\mathrm{d}\theta', \quad 0 \leqslant r < R, \tag{5-14}$$

自然积分方程为

$$u_n(\theta) = -\frac{1}{4\pi R}\int_0^{2\pi}\frac{u_0(\theta')}{\sin^2\dfrac{\theta - \theta'}{2}}\mathrm{d}\theta'. \tag{5-15}$$

半径为 R 的圆外区域调和边值问题的泊松积分公式为

$$u(r,\theta) = \frac{r^2 - R^2}{2\pi}\int_0^{2\pi}\frac{u_0(\theta')}{R^2 + r^2 - 2Rr\cos(\theta - \theta')}\mathrm{d}\theta', \quad r > R, \tag{5-16}$$

自然积分方程仍为(5-15)式.

关于这方面更多的结果请见文献[4].

5.3 超奇异积分方程的数值求解

自然边界归化导致超奇异积分方程.当 Ω 为圆内或圆外区域时,从调和方程、

重调和方程、平面弹性方程及斯托克斯方程等典型方程的边值问题归化得到的自然积分方程的积分核都含有超奇异项. 设边界上的基函数为 $L_i(\theta)$, $i=1,\cdots,N$, 必须计算

$$q_{ij}=\int_\Gamma\int_\Gamma\left(-\frac{1}{4\pi\sin^2\frac{\theta-\theta'}{2}}\right)L_j(\theta')L_i(\theta)\mathrm{d}\theta'\mathrm{d}\theta \tag{5-17}$$

形式的积分. 这一积分不能用通常的数值积分方法进行近似计算. 应用广义函数论中的重要公式

$$-\frac{1}{4\pi\sin^2\frac{\theta}{2}}=\frac{1}{2\pi}\sum_{n=-\infty}^{\infty}|n|\mathrm{e}^{in\theta}=\frac{1}{\pi}\sum_{n=1}^{\infty}n\cos n\theta, \tag{5-18}$$

则可得

$$q_{ij}=\frac{1}{\pi}\sum_{n=1}^{\infty}n\int_0^{2\pi}\int_0^{2\pi}\cos n(\theta-\theta')L_i(\theta)L_j(\theta')\mathrm{d}\theta'\mathrm{d}\theta. \tag{5-19}$$

特别当 $L_i(\theta)$, $i=1,\cdots,N$, 为分片线性基函数时, 经演算可得

$$q_{ij}=a_{|i-j|},\quad i,j=1,\cdots,N, \tag{5-20}$$

其中

$$a_k=\frac{4N^2}{\pi^3}\sum_{n=1}^{\infty}\frac{1}{n^3}\sin^4\frac{n\pi}{N}\cos\frac{nk}{N}2\pi,\quad k=0,1,\cdots,N-1, \tag{5-21}$$

这是一个收敛级数. 于是自然边界元的刚度矩阵便可得到. 例如对单位圆内调和方程, 自然边界元刚度矩阵即是由 $a_0,a_1,\cdots,a_{N-1}$ 生成的循环矩阵:

$$\boldsymbol{Q}=[q_{ij}]_{N\times N}=\begin{bmatrix} a_0 & a_1 & \cdots & a_{N-2} & a_{N-1}\\ a_{N-1} & a_0 & \cdots & a_{N-3} & a_{N-2}\\ \vdots & \vdots & & \vdots & \vdots\\ a_2 & a_3 & \cdots & a_0 & a_1\\ a_1 & a_2 & \cdots & a_{N-1} & a_0\end{bmatrix}, \tag{5-22}$$

且其中

$$a_i=a_{N-i},\quad i=1,\cdots,N-1,$$

故 $\boldsymbol{Q}$ 是对称循环矩阵.

对于分段二次或高次元, 也可得刚度矩阵系数的收敛级数表达式. 但对分段常数元则不然, 得到的系数表达式是发散级数, 这是因为分段常数基函数不属于 $H^{\frac{1}{2}}(\Gamma)$, 积分(5-17)没有意义.

上述积分核级数展开法当然并非是求解超奇异积分方程的唯一方法.

5.4 自然边界元与有限元耦合法

自然边界元方法和有限元方法基于同一变分原理, 且自然边界归化保持能量不变, 故自然边界元与有限元的耦合是自然而直接的. 可将求解区域分为两个子区

域,在一个有限的子区域上问题可以是非线性、非均匀的,而在另一个无限但规则的子区域上,问题则是线性的、均匀的,通常取圆周为两个子区域的交界线.可在前一子区域用有限元法,而在后一子区域用自然边界元法,二者耦合恰好取长补短.

设 Ω 为光滑闭曲线 Γ 的外部区域.考察 Ω 上的拉普拉斯方程的诺伊曼边值问题

$$\begin{cases}\Delta u = 0, & \Omega \text{ 内},\\ \dfrac{\partial u}{\partial n} = g, & \Gamma \text{ 上},\end{cases} \tag{5-23}$$

其中 $g \in H^{-\frac{1}{2}}(\Gamma)$ 并满足相容性条件,u 满足适当的无穷远条件.设

$$Q(u,v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}x,$$

则(5-23)等价于变分问题

$$\begin{cases}\text{求 } u \in W_0^1(\Omega), \text{使得}\\ Q(u,v) = \int_\Gamma gv\mathrm{d}s, \forall v \in W_0^1(\Omega).\end{cases} \tag{5-24}$$

无界区域 Ω 上的函数空间 $W_0^1(\Omega)$ 的定义见[4].直接由(5-24)式出发的离散化便导致有限元法,但由于 Ω 为无界区域,直接用有限元法求解难以得到满意的结果.今作半径为 R 的圆周 Γ'' 包围 Γ,Γ'' 为人工边界.无妨设其圆心为坐标原点,Γ'' 分 Ω 为有界子区域 Ω_1 及无界的圆外区域 Ω_2.于是有

$$\begin{aligned}Q(u,v) &= \int_{\Omega_1} \nabla u \cdot \nabla v\mathrm{d}x + \int_{\Omega_2} \nabla u \cdot \nabla v\mathrm{d}x\\ &= Q_1(u,v) + Q_2(u,v),\end{aligned} \tag{5-25}$$

其中

$$Q_i(u,v) = \int_{\Omega_i} \nabla u \cdot \nabla v\mathrm{d}x, \quad i = 1,2.$$

对区域 Ω_2 可应用自然边界归化得 Γ'' 上的自然积分方程如(5-15)式所示.令

$$\widehat{Q}_2(u_0,v_0) = \int_0^{2\pi}\int_0^{2\pi}\left(-\frac{1}{4\pi\sin^2\dfrac{\theta-\theta'}{2}}\right)u_0(\theta')v_0(\theta)\mathrm{d}\theta'\mathrm{d}\theta, \tag{5-26}$$

于是当 $\Delta u = 0$ 及 u,v 满足无穷远条件时,应用格林公式可得

$$Q_2(u,v) = \widehat{Q}_2(\gamma' u, \gamma' v), \tag{5-27}$$

其中 γ' 为 Ω 中函数到 Γ'' 上的迹算子.此即自然边界归化的能量不变性.于是变分问题(5-24)等价于有界子区域 Ω_1 上的变分问题

$$\begin{cases}\text{求 } u \in H^1(\Omega_1), \text{使得}\\ Q_1(u,v) + \widehat{Q}_2(\gamma' u, \gamma' v) = \int_\Gamma gv\mathrm{d}s, \quad \forall v \in H^1(\Omega_1).\end{cases} \tag{5-28}$$

其相应的离散问题为

$$\begin{cases}\text{求 } u_h \in S_h(\Omega_1), \text{使得}\\ Q_1(u_h,v_h) + \widehat{Q}_2(\gamma' u_h, \gamma' v_h) = \int_\Gamma gv_h\mathrm{d}s, \quad \forall v_h \in S_h(\Omega_1),\end{cases} \tag{5-29}$$

其中 $S_h(\Omega_1) \subset H^1(\Omega_1)$ 为 Ω_1 上的有限元解空间.

对 Ω_1 作有限元剖分,使得其在人工边界 Γ'' 上的节点为 Γ'' 的等分点.取分片线性基函数,由问题(5-29)出发可得线性代数方程组

$$QU = b, \tag{5-30}$$

其中 $Q = Q_1 + Q_2$, Q_1 可由通常的有限元法得到, Q_2 正是圆外区域的自然边界元刚度矩阵,它是由 $a_0, a_1, \cdots, a_{N-1}$ 所生成的循环矩阵,其中 $a_k, k = 1, \cdots, N-1$,由(5-21)式给出.

上述耦合法也可用于断裂及凹角区域.进一步的结果可见文献[4].

Γ'' 上的自然积分方程是人工边界上准确的边界条件,由此出发可导出一系列近似的微分边界条件或积分边界条件.在有界子区域内用有限元法求解时,加上这样的近似边界条件,自然可期望得到比简单地在人工边界上置零边界条件更好的结果.这是另一类较精确地处理无界区域问题的方法(见文献[4]).

6 自适应有限元边界元法

科学技术的发展对计算提出了越来越高的要求,仅仅进行一次计算得到一个近似解往往是不够的,还必须知道计算的精度及进一步改进计算的方法.也就是说,计算过程还应包括:对计算结果作后验误差估计;根据后验估计控制计算过程;通过后处理从近似解得到所需要的高精度数据.

6.1 自适应有限元法

20 世纪 70 年代末自适应技术首先被应用于有限元法中.有限元逼近通常分为 h 型、p 型及 h-p 型.h 型方法通过不断细分网格从而使单元尺寸 $h \to 0$ 来实现逼近.p 型方法通过不断提高分片多项式的次数使 $p \to \infty$ 来实现逼近.h-p 型方法则是二者的适当结合.对许多问题,通过细分全部网格或提高所有单元的插值多项式的次数来改善计算结果,显然很不经济且并非必要.自适应方法的特点在于依据对已有计算结果的后验误差估计,对网格进行局部细分或仅对某些单元提高插值多项式的次数,然后进行下一步计算.这样便可以尽可能少的代价取得尽可能好的结果.

由于自适应方法允许局部细分网格,故必须突破标准有限元法对网格的限制,即由正规网格推广到某种非正规网格,由拟一致网格推广到 K- 网格.

设计自适应过程的关键是选择可靠的误差指示值,因此获得可靠的可计算的后验局部误差估计是自适应过程设计的重要任务.

考察二阶椭圆边值问题

$$\begin{cases} \mathrm{L}u = -\sum_{k,l=1}^{2}\mathrm{D}_k(a_{kl}\mathrm{D}_l u) + bu = f, & \Omega\text{ 内}, \\ u = 0, & \Gamma_0\text{ 上}, \\ \sum_{k=1}^{2}\sigma_k(u)n_k = \sum_{k=1}^{2}\left(\sum_{l=1}^{2}a_{kl}\mathrm{D}_l u\right)n_k = g, & \Gamma_1\text{ 上} \end{cases} \tag{6-1}$$

及其变分形式

$$\begin{cases} \text{求 } u \in H_0^1(\Omega)\text{,使得} \\ B(u,v) = \int_\Omega fv\mathrm{d}x + \int_{\Gamma_1} gv\mathrm{d}s, \quad \forall v \in H_0^1(\Omega), \end{cases} \tag{6-2}$$

其中 $\mathrm{D}_l = \frac{\partial}{\partial x_l}, l = 1,2$,

$$B(u,v) = \int_\Omega \sum_{k,l=1}^{2}(a_{kl}\mathrm{D}_l u\mathrm{D}_k v + buv)\mathrm{d}x.$$

$$H_0^1(\Omega) = \{v \in H^1(\Omega) \mid v = 0, \Gamma_0\text{ 上}\}.$$

进行后验误差估计的最简单的途径是借用标准有限元法中已有的先验估计,如

$$\| u - u_h \|_{s,\Omega} \leqslant ch^{k-s}\| u \|_{k,\Omega}, \quad k > s, \tag{6-3}$$

但这一估计并非局部估计,且其右端为未知解函数 u 的高阶模,本身不可计算.问题化为如何由 u_h 计算 u 的高阶模的近似值.

Babuska 等人提出了通过剩余量进行后验估计的方法,特别对二阶椭圆边值问题及 K 网格给出了 $p = 1$ 时 h 型方法的如下结果:

$$\| u - u_h \|_{1,\Omega^0}^2 \leqslant c\left(h^2\sum_{\Delta}^{0}\| R \|_{0,\Delta}^2 + h\sum_{\Delta}^{0}\| r \|_{0,\partial\Delta}^2\right), \tag{6-4}$$

其中 $\Omega^0 \subset \Omega$ 为某个一致剖分子区域,$\sum_{\Delta}^{0}$ 表示对 Ω^0 内的单元求和,$R = \mathrm{L}u_h - f$ 为剩余量,r 为与近似解导数在单元边界上的跃度有关的"线剩余".这一估计是局部的.

更进一步,对正方形 K 网格上的双线性元,在一致网格片 Ω^0 上有渐近准确的后验局部误差估计:

$$\| u - u_h \|_{1,\Omega^0}^2 = \sum_{\Delta}^{0}\sum_{j=1}^{2}\sum_{k=1}^{4}\frac{|\Delta|^2}{48}[\mathrm{D}_j u_h(p_k)]^2 + \varepsilon, \tag{6-5}$$

其中 $|\Delta|$ 为单元边长,p_k 为单元顶点,$[\mathrm{D}_j u_h(p_k)]$ 为 p_k 点处 u_h 的 x_j 方向导数越过单元边界时的跃度,ε 为高阶小量.

关于任意双偶次元及双奇次元的相应的渐近准确后验局部误差估计可见[9]及其所引文献.

值得注意的是,对双奇次矩形元误差可仅用"线剩余"估计,而对双偶次矩形元误差只能用"面剩余"估计(见文献[10,11]).

6.2 自适应边界元法

将自适应技术应用于边界元计算始于80年代中.计算实践表明自适应方法对边界元计算同样有效.

在有限元法中,微分算子的局部性及有限元基函数仅有局部支集,保证了用局部剩余估计局部误差的有效性.而在边界元法中,由于边界归化得到的边界积分算子为整数阶的拟微分算子,有拟局部性质,及边界元基函数仅有局部支集,也保证了可用局部剩余来估计局部误差.

考察平面有界区域的充分光滑边界 Γ 上的积分方程

$$\mathrm{A}u = \int_\Gamma a(s,t)u(t)\mathrm{d}t = f(s), \tag{6-6}$$

其中 $\mathrm{A}: H^\gamma(\Gamma) \to H^{-\gamma}(\Gamma)$ 为 Γ 上 $\alpha = 2\gamma$ 阶强椭圆拟微分算子.假定方程(6-6)有唯一解,定义双线性型

$$\mathrm{A}(u,v) = (\mathrm{A}u,v) = \int_\Gamma v\mathrm{A}u\mathrm{d}s.$$

则积分方程等价于变分问题

$$\begin{cases} \text{求 } u \in H^\gamma(\Gamma), \text{使得} \\ \mathrm{A}(u,v) = \int_\Gamma fv\mathrm{d}s, \forall v \in H^\gamma(\Gamma). \end{cases} \tag{6-7}$$

其近似变分问题为

$$\begin{cases} \text{求 } u_h \in S_h, \text{使得} \\ \mathrm{A}(u_h,v_h) = \int_\Gamma fv_h\mathrm{d}s, \quad \forall v_h \in S_h. \end{cases} \tag{6-8}$$

若逼近函数空间 $S_h \subset H^k(\Gamma)$,满足逼近定理

$$\inf_{v_h \in S_h} \| v - v_h \|_j \leqslant ch^{m-j} \| v \|_m, \quad \forall v \in H^{m+1}(\Gamma),$$

$0 \leqslant j \leqslant m, j \leqslant k$,则误差满足如下先验估计:

$$\| u - u_h \|_l \leqslant ch^{m-l} \| u \|_m, \tag{6-9}$$

$l \leqslant \gamma \leqslant m, \gamma \leqslant k$,而剩余量

$$R = \mathrm{A}(u - u_h) = f - Au_h. \tag{6-10}$$

当近似解 u_h 求得后,剩余量 R 是可计算的.

先验估计(6-9)式是整体误差估计,并未提供关于局部误差分布的信息.下面的结果是通过分析剩余量在 Γ 上的分布来估计误差的分布.

若 A 为 α 阶强椭圆拟微分算子,边界元定义在 K- 网格族上,$u \in H^k(I_2) \cap H^l(\Gamma)$, $I_1 \subset \mathring{I}_2 \subset \Gamma$,则 $S_h = S_h^{k,t} \subset H^k(\Gamma)$ 上的 h 型边界元解满足如下后验局部误差估计(见[9]):

当 $\alpha = -1$ 时,

$$\| u - u_h \|_{0,I_1} \leqslant c(\| R \|_{1,I_2} + h^{k+1+l} \| u \|_{l,\Gamma} + h^k \| R \|_{1,\Gamma}), \tag{6-11}$$

当 $\alpha = 0$ 时,

$$\| u - u_h \|_{0,I_1} \leqslant c(\| R \|_{0,I_2} + h^{k+1+l} \| u \|_{l,\Gamma} + h^k \| R \|_{0,\Gamma}), \quad (6\text{-}12)$$

当 $\alpha = 1$ 时,

$$\| u - u_h \|_{0,I_1} \leqslant c(h \| R \|_{0,I_2} + h^{k+1+l} \| u \|_{l,\Gamma} + h^{k+1} \| R \|_{0,\Gamma}). \quad (6\text{-}13)$$

在大多数应用中,边界积分算子都属于 $\alpha = -1$(弱奇异积分方程)、$\alpha = 0$(柯西型奇异积分方程)及 $\alpha = 1$(超奇异积分方程)这三种情况. 于是由(6-11)等式可见,剩余量适当的局部范数可以作为误差指示值.

对于 p 型边界元方法,局部误差满足如下估计:

$$\begin{cases} \| e \|_{\alpha,I_1} \leqslant c_1 \| R \|_{0,I_2} + c_2 p^{-1} \| e \|_{\alpha,\Gamma}, \\ \| e \|_{\alpha,I_2} \geqslant c_3 \| R \|_{0,I_1} - c_4 p^{-1} \| e \|_{\alpha,\Gamma}, \end{cases} \quad \alpha = 0,1, \quad (6\text{-}14)$$

及

$$\begin{cases} \| e \|_{0,I_1} \leqslant c_1 \| R \|_{1,I_2} + c_2 p^{-\frac{1}{2}} \| e \|_{-\frac{1}{2},\Gamma}, \\ \| e \|_{0,I_2} \geqslant c_3 \| R \|_{1,I_1} - c_4 p^{-\frac{1}{2}} \| e \|_{-\frac{1}{2},\Gamma}, \end{cases} \quad \alpha = -1, \quad (6\text{-}15)$$

其中 e 为误差. 于是对 p 型边界元法,近似解的局部误差也可用剩余量的局部模作渐近估计.

7 区域分解算法

区域分解算法是20世纪80年代以来获得迅速发展的偏微分方程数值求解新技术,该类算法把计算区域分解为若干子区域,将原问题的求解转化为定义在子区域上的一系列简单问题的求解,从而将问题由大化小,由复杂化简单,且便于进行并行计算. 由于允许在不同子区域建立不同的数学模型,选择不同的计算方法,进行不同的网格剖分,应用现有的各种软件,特别是可以在若干规则的子区域上采用快速傅里叶变换、自然边界元方法、谱方法等快速高效算法,这类算法与其它方法相比有特别的灵活性和显著的优越性.

7.1 有限元区域分解算法

有限元区域分解算法将有界的求解区域分为两个或多个重叠的或不重叠的子区域,在各子区域用有限元法求解.

施瓦兹(Schwarz)交替算法便是一类重叠型区域分解算法. 考虑如下模型问题:

$$\begin{cases} -\Delta u = f, & \Omega \text{ 内}, \\ u = 0, & \partial\Omega \text{ 上}. \end{cases} \quad (7\text{-}1)$$

将 Ω 分为两个重叠的子区域 Ω_1 及 Ω_2. 首先选择初始 $u^{(0)} \in H_0^1(\Omega)$. 令 $u^{(2n+1)}$、

$u^{(2n+2)}, n = 0,1,\cdots$,分别满足子区域上的方程

$$\begin{cases} -\Delta u^{(2n+1)} = f, & \Omega_1 \text{ 内}, \\ u^{(2n+1)} = u^{(2n)}, & \partial\Omega_1 \text{ 上}, \end{cases} \tag{7-2}$$

及

$$\begin{cases} -\Delta u^{(2n+2)} = f, & \Omega_2 \text{ 内}, \\ u^{(2n+2)} = u^{(2n+1)}, & \partial\Omega_2 \text{ 上}. \end{cases} \tag{7-3}$$

交替求解方程(7-2)及(7-3).令 $V_i = H_0^1(\Omega_i), i = 1,2$,用 $e^{(k)} = u - u^{(k)}$ 表示误差.于是有如下结果.

定理 1 如果 $V_1^{\perp} \cap V_2^{\perp} = \{0\}$,或等价地成立 $V = \overline{V_1 + V_2}$,则 $e^{(n)} \to 0(n \to \infty)$.若 $V = V_1 + V_2$,则必存在常数 $\alpha \in [0,1)$,使得

$$\| e^{(n+2)} \|_1 \leqslant \alpha \| e^{(n)} \|_1, \tag{7-4}$$

即收敛是几何的.

狄利克雷问题(7-2)及(7-3)均可用标准有限元法求解.

施瓦兹交替法可推广到多子区域的情况.

在不重叠区域分解的情况采用 D-N 交替法(即狄利克雷 - 诺伊曼方法),斯蒂克洛夫 - 庞加莱(Steklov-Poincare)算子在其中起关键作用.

考虑非齐次边值问题

$$\begin{cases} \mathrm{L}u = f, & \Omega \text{ 内}, \\ u = g, & \partial\Omega \text{ 上}. \end{cases} \tag{7-5}$$

将 Ω 分为不重叠的子区域 Ω_1 及 Ω_2,$\Gamma = \partial\Omega_1 \cap \partial\Omega_2$.构造 **D-N 交替算法**如下:

步 1 选初始 $\lambda^{(0)} \in \Phi = \{v|_{\Gamma} \mid v \in H_0^1(\Omega)\}$,置 $n := 0$.

步 2 在 Ω_1 上解狄利克雷问题

$$\begin{cases} \mathrm{L}u_1^{(n)} = f, & \Omega_1 \text{ 内}, \\ u_1^{(n)} = \lambda^{(n)}, & \Gamma \text{ 上}, \\ u_1^{(n)} = g, & \partial\Omega_1 \setminus \Gamma \text{ 上}. \end{cases} \tag{7-6}$$

步 3 在 Ω_2 上解诺伊曼问题

$$\begin{cases} \mathrm{L}u_2^{(n)} = f, & \Omega_2 \text{ 内}, \\ \dfrac{\partial u_2^{(n)}}{\partial n_2} = \dfrac{\partial u_1^{(n)}}{\partial n_2}, & \Gamma \text{ 上}, \\ u_2^{(n)} = g, & \partial\Omega_2 \setminus \Gamma \text{ 上}. \end{cases} \tag{7-7}$$

步 4 计算或输入 θ_n,并置

$$\lambda^{(n+1)} = \theta_n u_2^{(n)} + (1 - \theta_n)\lambda^{(n)}, \quad \Gamma \text{ 上}. \tag{7-8}$$

步 5 置 $n := n + 1$ 转第 2 步.

定理 2 D-N 交替法与预处理理查德森(Richardson)迭代法

$$S_2(\lambda^{(n+1)} - \lambda^{(n)}) = \theta_n(\chi - S\lambda^{(n)}) \tag{7-9}$$

等价,其中 $S = S_1 + S_2$ 为斯蒂克洛夫 - 庞加莱算子,$S_k = \dfrac{\partial}{\partial n_k}(\mathrm{R}_k \cdot), k = 1,2, \mathrm{R}_k$

为调和扩张算子，$\chi=\frac{\partial}{\partial n_1}(\mathrm{T}_2-\mathrm{T}_1)f$，$\mathrm{T}_k f$满足

$$\begin{cases}\mathrm{LT}_k f=f, & \Omega_k \text{内}, \\ \mathrm{T}_k f=0, & \Gamma \text{上}, \\ \mathrm{T}_k f=g, & \partial\Omega_k \setminus \Gamma \text{上}. k=1,2.\end{cases}$$

定理3 若选定参数

$$\theta=\frac{2}{2+T^{-1}+\sigma},$$

则迭代(7-9)收敛，且收敛速度为$\frac{2(1+T^{-1})}{1+\sigma}$，其中

$$\sigma=\sup_{\lambda\in\Phi}\frac{\| R_1\lambda \|_{(1)}^2}{\| R_2\lambda \|_{(2)}^2},\quad T=\sup_{\lambda\in\Phi}\frac{\| R_2\lambda \|_{(2)}^2}{\| R_1\lambda \|_{(1)}^2}. \tag{7-10}$$

在D-N算法的第2步及第3步中应用有限元法求解问题(7-6)及(7-7)，便导致离散D-N交替法.

定理4 离散D-N交替法的迭代矩阵$[S_h^{(2)}]^{-1}S_h$的条件数及收敛速度与有限元网格参数h无关.

7.2 基于边界归化的区域分解算法

有限元方法及其区域分解算法尽管对有界区域问题很有效，但对无界区域问题则难以获得理想的精度.边界归化是求解无界区域问题的强有力的手段.自20世纪70年代以来，有限元与边界元的耦合已成为求解无界区域问题的主要方法，尤其是自然边界元与有限元耦合更有许多独特的优点.但耦合法的刚度矩阵已不再是带状稀疏的，已有的有限元程序也不能被直接应用.

将边界归化与区域分解相结合，导致了基于边界归化的区域分解算法，这类算法特别适于求解无界区域问题.这是与国际流行的有限元区域分解算法完全不同的一类新型区域分解算法.

考察调和方程的外边值问题

$$\begin{cases}-\Delta w=0, & \Omega \text{内}, \\ w=g, & \Gamma_0 \text{上},\end{cases} \tag{7-11}$$

其中Ω为闭曲线Γ_0的外部区域.此问题等价于泊松方程的齐次边值问题

$$\begin{cases}-\Delta u=f, & \Omega \text{内}, \\ u=0, & \Gamma_0 \text{上}.\end{cases} \tag{7-12}$$

作半径为R_1及R_2的同心圆周Γ_1及Γ_2包围Γ_0，$R_1>R_2>0$.设Ω_1为Γ_0与Γ_1间的有界区域，Ω_2为Γ_2外部的无界区域.定义如下施瓦兹交替算法：

$$\begin{cases}-\Delta u_1^{(k)}=f, & \Omega_1 \text{内}, \\ u_1^{(k)}=0, & \Gamma_0 \text{上}, \qquad k=1,2,\cdots \\ u_1^{(k)}=u_2^{(k-1)}, & \Gamma_1 \text{上},\end{cases} \tag{7-13}$$

及

$$\begin{cases} -\Delta u_2^{(k)} = f, & \Omega_2 \text{ 内}, \\ u_2^{(k)} = u_1^{(k)}, & \Gamma_2 \text{ 上}, \end{cases} \quad k = 1,2,\cdots, \tag{7-14}$$

其中 $u_2^{(0)} \in H^{\frac{1}{2}}(\Gamma_1)$ 任意给定.

用 $e_i^{(k)} = u - u_i^{(k)}, i = 1,2$,表示误差,则有

定理 5 施瓦兹交替法(7-13)及(7-14)几何收敛:

$$\lim_{k\to\infty} \| e_i^{(k)} \|_1 = 0, \quad i = 1,2, \tag{7-15}$$

且存在常数 $\alpha \in [0,1)$,使得

$$\| e_1^{(k)} \|_1 \leqslant \alpha^{k-1} \| e_1^{(1)} \|_1, \quad \| e_2^{(k)} \|_1 \leqslant \alpha^k \| e_2^{(0)} \|_1. \tag{7-16}$$

在实际计算中,在 Ω_1 上可用标准有限元法,而在 Ω_2 上则可直接应用通过自然边界归化得到的泊松积分公式.

仍考察 Γ_0 外部区域 Ω 上的泊松方程的狄利克雷问题

$$\begin{cases} -\Delta u = f, & \Omega \text{ 内}, \\ u = g, & \Gamma_0 \text{ 上}. \end{cases} \tag{7-17}$$

附加无穷远条件使之存在唯一解.作半径为 R_1 的圆周 Γ_1 包围 Γ_0,则人工边界 Γ_1 分 Ω 为内子区域 Ω_1 及外子区域 Ω_2.定义如下 D-N 区域分解算法:

步 1 选初始 $\lambda^{(0)} \in H^{\frac{1}{2}}(\Gamma_1)$, $n := 0$.

步 2 在 Ω_2 上解狄利克雷问题

$$\begin{cases} -\Delta u_2^{(n)} = f, & \Omega_2 \text{ 内}, \\ u_2^{(n)} = \lambda^{(n)}, & \Gamma_1 \text{ 上}. \end{cases} \tag{7-18}$$

步 3 在 Ω_1 上解边值问题

$$\begin{cases} -\Delta u_1^{(n)} = f, & \Omega_1 \text{ 内}, \\ \dfrac{\partial u_1^{(n)}}{\partial n_1} = -\dfrac{\partial u_2^{(n)}}{\partial n_2}, & \Gamma_1 \text{ 上}, \\ u_1^{(n)} = g, & \Gamma_0 \text{ 上}. \end{cases} \tag{7-19}$$

步 4 置

$$\lambda^{(n+1)} = \theta_n u_1^{(n)} + (1-\theta_n)\lambda^{(n)}, \quad \Gamma_1 \text{ 上}. \tag{7-20}$$

步 5 令 $n := n+1$,转第 2 步.

注意到第 2 步为求解圆外区域的狄利克雷问题,而第 3 步中仅需要问题(7-18)之解在 Γ_1 上的法向导数,于是根本不必求解问题(7-18),而只要应用[4]中给出的自然积分方程直接由 $\lambda^{(n)}$ 求 $\dfrac{\partial u_2^{(n)}}{\partial n_2}$ 即可.这是一个超奇异积分计算问题,工作量很小.

定理 6 当 $0 < \min\theta_n \leqslant \max\theta_n < 1$ 时,离散 D-N 交替法收敛,且收敛速度与迭代矩阵的条件数均与有限元网格参数 h 无关.

特别取 $\theta_n = \dfrac{2}{3}$ 时,迭代收缩因子必不大于 $\dfrac{1}{3}$.

基于边界归化的区域分解算法将区域分解算法的适用范围拓广到无界区域问题,尤其由于自然边界归化的独特优点,使得这一方法更加简便易行.本方法只要在较小的有界区域内直接应用有限元标准程序,而涉及自然边界归化的计算量又极小,因此显著减少了计算量.

参考文献

1 冯康.基于变分原理的差分格式.应用数学与计算数学,1965,2:4,238 ~ 262.

2 冯康,石钟慈.弹性结构的数学理论.北京:科学出版社,1981.

3 应隆安.无限元方法.北京:北京大学出版社,1992.

4 余德浩.自然边界元方法的数学理论.北京:科学出版社,1993.

5 祝家麟.椭圆边值问题的边界元分析,北京:科学出版社,1991.

6 Shi Z G.The F-E-M-Test for convergence of nonconforming finite elements,Math of Comp,1987,49,391 ~ 405

7 Yu Dehao.Natural boundary element method and adaptive boundary element method——Some new developments in BEM in China.In:Contemporary Mathematics 163.In:American Mathematical Society,1994.185 ~ 204

8 Yu Dehao.Domain decomposition methods for unbounded domains.In:Glowinski R et al eds.Domain decomposition methods in science and engineering,8th Inter Conference.New York:John Wiley & Sons,1997.125 ~ 132

9 Yu Dehao.Adaptive methods in finite and boundary element computations.Progress in Natural Science,1994,4(2):137 ~ 146

10 余德浩.双偶次有限元的渐近准确误差估计.计算数学,1991,13(1):89 ~ 101.

11 余德浩,双奇次有限元的渐近准确误差估计,计算数学,1991,13(3):307 ~ 314.

12 余德浩,无界区域上基于自然边界归化的一种区域分解算法,计算数学,1994,16(4):448 ~ 459

13 余德浩,无界区域非重叠区域分解算法的离散化及其收敛性,计算数学,1996,18(3):328 ~ 336

·计算机数学卷·

第4篇

计算流体力学中的差分法

编　者　苏铭德

审校者　刘秋生

目　录

引　　言

随着计算机容量和速度的迅速发展,使越来越复杂的流场用数值模拟的方法进行研究成为可能.因此计算流体力学作为一门新兴的学科越来越得到广泛的应用.

由于流体力学的基本方程是强非线性的,尽管有限元法的理论和实践都已相当成熟,但在计算流体力学中大量采用的仍然是有限差分法.近年来由于非结构网格的出现,它与有限差分法相结合,使计算流体力学得到更加广泛的应用.

流体力学中的有限差分法的主要发展方向有以下两个方面:在高速流动中,出现激波,当流动非常复杂时,激波的结构将是很复杂的.因此为准确地捕捉和分辨激波,人们发展了各种有限差分格式,其中有 TVD, ENO, NND, BGK 等行之有效的重要格式.另一方面,尽管计算机已得到飞速的发展,但仍然赶不上人们对于像湍流这样复杂流动的数值模拟的需要,所以,为尽可能提高计算机的使用效率,发展高精度格式就成了另一个重要的方向.

流体力学范围非常广泛,本篇仅限于单相的气体或液体流动.介绍的差分格式也限于最重要的和最常用的.由于流体力学基本方程是强非线性的,流动十分复杂,目前尚无成熟的有关流体力学中有限差分法的理论.本文介绍的仅仅是有关模型方程的理论或仅有启发性的理论供读者参考.

1　流体力学的基本方程

1.1　守恒型和非守恒型方程组

流体运动遵循质量守恒、动量守恒和能量守恒定律.流体力学基本方程就是这些基本定律在描写流体运动时的数学模型,通常用守恒型式方程加以描述.

根据以上三个基本定律,流体力学基本方程的积分形式为

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int \rho \mathrm{d}\tau = 0, \\ \dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int \rho \boldsymbol{V} \mathrm{d}\tau = -\oint p\boldsymbol{n}\mathrm{d}\sigma + \oint \boldsymbol{\tau}_n \mathrm{d}\sigma + \int \rho \boldsymbol{f} \mathrm{d}\tau, \\ \dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int \rho\left(e + \frac{V^2}{2}\right)\mathrm{d}\tau = -\oint p v_n \mathrm{d}\sigma + \oint \boldsymbol{\tau}_n \cdot \boldsymbol{V}\mathrm{d}\sigma + \int \rho \boldsymbol{f} \cdot \boldsymbol{V}\mathrm{d}\tau + \\ \qquad\qquad \displaystyle\oint \lambda \frac{\partial T}{\partial n}\mathrm{d}\sigma + \int \rho q \mathrm{d}\tau, \end{cases} \tag{1-1}$$

其中 ρ 为密度，V 为速度向量，$\tau_n = \tau \cdot n$，$\sigma = -pI + \tau$，τ 为切应力张量，n 为封闭曲面外法线单位向量，e 为内能，T 为温度，λ 为导热系数，f 为外界激体力，q 为外界激体加热，$d\tau$ 为体积微元，$d\sigma$ 为积分面元.

利用高斯(Guass)积分公式并计及积分体积的任意性，可以得到守恒型方程组

$$
\begin{cases}
\dfrac{\partial\rho}{\partial t} + \dfrac{\partial\rho u}{\partial x} + \dfrac{\partial\rho v}{\partial y} + \dfrac{\partial\rho w}{\partial z} = 0, \\
\dfrac{\partial\rho u}{\partial t} + \dfrac{\partial\rho uu}{\partial x} + \dfrac{\partial\rho uv}{\partial y} + \dfrac{\partial\rho uw}{\partial z} = \rho f_x - \dfrac{\partial p}{\partial x} + \dfrac{\partial\tau_{xx}}{\partial x} + \dfrac{\partial\tau_{xy}}{\partial y} + \dfrac{\partial\tau_{xz}}{\partial z}, \\
\dfrac{\partial\rho v}{\partial t} + \dfrac{\partial\rho uv}{\partial x} + \dfrac{\partial\rho vv}{\partial y} + \dfrac{\partial\rho vw}{\partial z} = \rho f_y - \dfrac{\partial p}{\partial y} + \dfrac{\partial\tau_{yx}}{\partial x} + \dfrac{\partial\tau_{yy}}{\partial y} + \dfrac{\partial\tau_{yz}}{\partial z}, \\
\dfrac{\partial\rho w}{\partial t} + \dfrac{\partial\rho uw}{\partial x} + \dfrac{\partial\rho vw}{\partial y} + \dfrac{\partial\rho ww}{\partial z} = \rho f_z - \dfrac{\partial p}{\partial z} + \dfrac{\partial\tau_{zx}}{\partial x} + \dfrac{\partial\tau_{zy}}{\partial y} + \dfrac{\partial\tau_{zz}}{\partial z}, \\
\dfrac{\partial}{\partial t}[\rho(e + \dfrac{V^2}{2})] + \dfrac{\partial}{\partial x}[\rho u(e + \dfrac{V^2}{2})] + \dfrac{\partial}{\partial y}[\rho v(e + \dfrac{V^2}{2})] + \dfrac{\partial}{\partial z}[\rho w(e + \dfrac{V^2}{2})] \\
= \rho f \cdot V - \dfrac{\partial pu}{\partial x} - \dfrac{\partial pv}{\partial y} - \dfrac{\partial pw}{\partial z} + \\
\quad \dfrac{\partial}{\partial x}(u\tau_{xx} + v\tau_{yx} + w\tau_{zx}) + \dfrac{\partial}{\partial y}(u\tau_{xy} + v\tau_{yy} + w\tau_{zy}) + \\
\quad \dfrac{\partial}{\partial z}(u\tau_{xz} + v\tau_{yz} + w\tau_{zz}) + \\
\quad \dfrac{\partial}{\partial x}(\lambda\dfrac{\partial T}{\partial x}) + \dfrac{\partial}{\partial y}(\lambda\dfrac{\partial T}{\partial y}) + \dfrac{\partial}{\partial z}(\lambda\dfrac{\partial T}{\partial z}) + \rho q.
\end{cases}
\tag{1-2}
$$

当流体为常比热完全气体时，有 $p = \rho RT$，$e = C_V T$，C_V 为定容比热容，R 为气体常数.

不计激体力和激体加热，上述方程组可改写为向量形式，

$$
\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = \frac{\partial F_V}{\partial x} + \frac{\partial G_V}{\partial y} + \frac{\partial H_V}{\partial z}, \tag{1-3}
$$

其中

$$
\begin{cases}
U = (\rho, \rho u, \rho v, \rho w, \rho E_t)^T, \\
F = (\rho u, \rho u^2 + p, \rho uv, \rho uw, u(\rho E_t + p))^T, \\
F_V = (0, \tau_{xx}, \tau_{yx}, \tau_{zx}, \lambda\dfrac{\partial T}{\partial x} + u\tau_{xx} + v\tau_{yx} + w\tau_{zx})^T, \\
G = (\rho v, \rho uv, \rho v^2 + p, \rho vw, v(E_t + p))^T, \\
G_V = (0, \tau_{xy}, \tau_{yy}, \tau_{zy}, \lambda\dfrac{\partial T}{\partial y} + u\tau_{xy} + v\tau_{yy} + w\tau_{zy})^T, \\
H = (\rho w, \rho uw, \rho vw, \rho w^2 + p, w(E_t + p))^T, \\
H_V = (0, \tau_{xz}, \tau_{yz}, \tau_{zz}, \lambda\dfrac{\partial T}{\partial z} + u\tau_{xz} + v\tau_{yz} + w\tau_{zz})^T,
\end{cases}
\tag{1-4}
$$

和

$$\rho E_s = \rho(e + \frac{V^2}{2}). \tag{1-5}$$

由本构方程知

$$\boldsymbol{\tau} = (\mu' - \frac{2}{3}\mu)(\nabla \cdot \boldsymbol{V})\boldsymbol{I} + 2\mu \boldsymbol{S}, \tag{1-6}$$

$$\boldsymbol{S} = [S_{ij}] = [\frac{1}{2}(\frac{\partial v_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i})], \tag{1-7}$$

其中 μ'、μ 分别为流体体膨胀粘性系数和粘性系数.

(1-3) 式可以改写为非守恒型. 首先定义

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} = \boldsymbol{A}, \quad \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} = \boldsymbol{B}, \quad \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{U}} = \boldsymbol{C}.$$

通过直接验算可知

$$\boldsymbol{AU} = \boldsymbol{F}, \quad \boldsymbol{BU} = \boldsymbol{H}, \quad \boldsymbol{CU} = \boldsymbol{H}.$$

其中

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{\gamma-1}{2}V^2 - u^2 & (3-\gamma)u & -(\gamma-1)v & -(\gamma-1)w & \gamma-1 \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ -\gamma uE_s + (\gamma-1)uV^2 & \gamma E_s - \frac{\gamma-1}{2}\rho(3u^2+v^2+w^2) & -(\gamma-1)uv & -(\gamma-1)uw & \gamma u \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ \frac{\gamma-1}{2}V^2 - v^2 & -(\gamma-1)v & (3-\gamma)v & -(\gamma-1)w & \gamma-1 \\ -vw & 0 & w & v & 0 \\ -\gamma vE_s + (\gamma-1)vV^2 & -(\gamma-1)uv & \gamma E_s - \frac{\gamma-1}{2}(u^2+3v^2+w^2) & -(\gamma-1)vw & \gamma v \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -uw & w & 0 & u & 0 \\ -vw & 0 & w & v & 0 \\ \frac{\gamma-1}{2}V^2 - w^2 & -(\gamma-1) & -(\gamma-1)v & (3-\gamma)w & \gamma-1 \\ -\gamma wE_s + (\gamma-1)wV^2 & -(\gamma-1)wu & -(\gamma-1)vw & \gamma E_s - \frac{\gamma-1}{2}(u^2+v^2+3w^2) & \gamma w \end{bmatrix}.$$

于是非守恒型方程可写作

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x} + \boldsymbol{B}\frac{\partial \boldsymbol{U}}{\partial y} + \boldsymbol{C}\frac{\partial \boldsymbol{U}}{\partial z} = \frac{\partial \boldsymbol{F}_v}{\partial x} + \frac{\partial \boldsymbol{G}_v}{\partial y} + \frac{\partial \boldsymbol{H}_V}{\partial z}. \tag{1-8}$$

不难分析知 $\boldsymbol{A}$ 的特征值 λ 为

$$\lambda = u + c, u, u, u, u - c. \tag{1-9}$$

记 $\boldsymbol{\Lambda}_A = \mathrm{diag}(u + c, u, u, u, u - c)$, 可得

$$A = S^{-1}\Lambda_A S, \tag{1-10}$$

其中

$$S^{-1} = \begin{bmatrix} 1 & \dfrac{1}{2c^2} & 0 & 0 & \dfrac{1}{2c^2} \\ u & \dfrac{1}{2c}\left(\dfrac{u}{c}+1\right) & 0 & 0 & \dfrac{1}{2c}\left(\dfrac{u}{c}-1\right) \\ v & \dfrac{v}{2c^2} & \rho & 0 & \dfrac{v}{2c^2} \\ w & \dfrac{w}{2c^2} & 0 & \rho & \dfrac{w}{2c^2} \\ \dfrac{V^2}{2} & \dfrac{1}{2}\left(\dfrac{V^2}{2c^2}+\dfrac{u}{c}+\dfrac{1}{\gamma-1}\right) & \rho v & \rho w & \dfrac{1}{2}\left(\dfrac{V^2}{2c^2}-\dfrac{u}{c}+\dfrac{1}{\gamma-1}\right) \end{bmatrix},$$

$$S = \begin{bmatrix} 1-\dfrac{\gamma-1}{2c^2}V^2 & (\gamma-1)\dfrac{v}{c^2} & (\gamma-1)\dfrac{V}{c^2} & (\gamma-1)\dfrac{w}{c^2} & -(\gamma-1)\dfrac{1}{c^2} \\ -uc+\dfrac{\gamma-1}{2}V^2 & c-(\gamma-1)u & -(\gamma-1)v & -(\gamma-1)w & \gamma-1 \\ -\dfrac{V}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{w}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ uc+\dfrac{\gamma-1}{2}V^2 & -c-(\gamma-1)u & -(\gamma-1)v & -(\gamma-1)w & \gamma-1 \end{bmatrix}.$$

类似地 B 的特征值 μ 为

$$\mu = v+c, v, v, v, v-c. \tag{1-11}$$

记

$$\Lambda_B = \mathrm{diag}(v, v+c, v, v, v-c),$$

有

$$B = T^{-1}\Lambda_B T, \tag{1-12}$$

其中

$$T^{-1} = \begin{bmatrix} 1 & \dfrac{1}{2c^2} & 0 & 0 & \dfrac{1}{2c^2} \\ u & \dfrac{u}{2c^2} & \rho & 0 & \dfrac{u}{2c^2} \\ v & \dfrac{1}{2c}\left(\dfrac{v}{c}+1\right) & 0 & 0 & \dfrac{1}{2c}\left(\dfrac{v}{c}+1\right) \\ w & \dfrac{w}{2c^2} & 0 & \rho & \dfrac{w}{2c^2} \\ \dfrac{V^2}{2} & \dfrac{1}{2}\left(\dfrac{V^2}{2c^2}+\dfrac{v}{c}+\dfrac{1}{\gamma-1}\right) & \rho u & \rho w & \dfrac{1}{2}\left(\dfrac{1}{2c^2}-\dfrac{v}{c}+\dfrac{1}{\gamma-1}\right) \end{bmatrix},$$

$$
\boldsymbol{T}=\begin{bmatrix}
1-\dfrac{\gamma-1}{2c^2}V^2 & (\gamma-1)\dfrac{u}{c^2} & (\gamma-1)\dfrac{v}{c^2} & (\gamma-1)\dfrac{w}{c^2} & -(\gamma-1)\dfrac{1}{c^2}\\
-vc+\dfrac{\gamma-1}{2}V^2 & -(\gamma-1)u & c-(\gamma-1)v & -(\gamma-1)w & \gamma-1\\
-\dfrac{u}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0\\
-\dfrac{w}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0\\
vc+\dfrac{\gamma-1}{2}V^2 & -(\gamma-1)u & -c-(\gamma-1)v & -(\gamma-1)w & \gamma-1
\end{bmatrix},
$$

以及 $\boldsymbol{C}$ 的特征值 ζ 为

$$\zeta = w+c, w, w, w, w-c. \tag{1-13}$$

记 $$\boldsymbol{\Lambda}_C=\mathrm{diag}(w, w+c, w, w, w-c),\text{有 } \boldsymbol{C}=\boldsymbol{Q}^{-1}\boldsymbol{\Lambda}_C\boldsymbol{Q}, \tag{1-14}$$

其中

$$
\boldsymbol{Q}^{-1}=\begin{bmatrix}
1 & \dfrac{1}{2c^2} & 0 & 0 & \dfrac{1}{2c^2}\\
u & \dfrac{u}{2c^2} & \rho & 0 & \dfrac{u}{2c^2}\\
v & \dfrac{v}{2c^2} & 0 & \rho & \dfrac{v}{2c^2}\\
w & \dfrac{1}{2c}\left(\dfrac{w}{c}+1\right) & 0 & 0 & \dfrac{1}{2c}\left(\dfrac{w}{c}-1\right)\\
\dfrac{V^2}{2} & \dfrac{1}{2}\left(\dfrac{V^2}{2c^2}+\dfrac{w}{c}+\dfrac{1}{\gamma-1}\right) & \rho u & \rho v & \dfrac{1}{2}\left(\dfrac{v^2}{2c^2}-\dfrac{w}{c}+\dfrac{1}{\gamma-1}\right)
\end{bmatrix},
$$

$$
\boldsymbol{Q}=\begin{bmatrix}
1-\dfrac{\gamma-1}{2c^2}V^2 & (\gamma-1)\dfrac{u}{c^2} & (\gamma-1)\dfrac{v}{c^2} & (\gamma-1)\dfrac{w}{c^2} & -(\gamma-1)\dfrac{1}{c^2}\\
-wc+\dfrac{\gamma-1}{2}V^2 & -(\gamma-1)u & -(\gamma-1)v & c-(\gamma-1)w & \gamma-1\\
-\dfrac{u}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0\\
-\dfrac{v}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0\\
wc+\dfrac{\gamma-1}{2}V^2 & -(\gamma-1)u & -(\gamma-1)v & -c-(\gamma-1)w & \gamma-1
\end{bmatrix}.
$$

设 $$\boldsymbol{W}=(\rho, u, v, w, p)^{\mathrm{T}}, \tag{1-15}$$

又记 $$\boldsymbol{D}^{-1}=\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{W}}, \tag{1-16}$$

则由(1-8)式得

$$\boldsymbol{D}^{-1}\frac{\partial \boldsymbol{W}}{\partial t}+\boldsymbol{A}\boldsymbol{D}^{-1}\frac{\partial \boldsymbol{W}}{\partial x}+\boldsymbol{B}\boldsymbol{D}^{-1}\frac{\partial \boldsymbol{W}}{\partial y}+\boldsymbol{C}\boldsymbol{D}^{-1}\frac{\partial \boldsymbol{W}}{\partial z}=\frac{\partial \boldsymbol{F}_V}{\partial x}+\frac{\partial \boldsymbol{G}_V}{\partial y}+\frac{\partial \boldsymbol{H}_V}{\partial z}.$$

两侧同时左乘 D 得

$$\frac{\partial W}{\partial t}+A_1\frac{\partial W}{\partial x}+B_1\frac{\partial W}{\partial y}+C_1\frac{\partial W}{\partial z}=D\frac{\partial F_V}{\partial x}+D\frac{\partial G_V}{\partial y}+D\frac{\partial H_V}{\partial z},\tag{1-17}$$

其中
$$A_1=DAD^{-1},\ B_1=DBD^{-1},\ C_1=DCD^{-1},\tag{1-18}$$
且有

$$A_1=\begin{bmatrix}u&\rho&0&0&0\\0&u&0&0&\gamma/\rho\\0&0&u&0&0\\0&0&0&u&0\\0&p&0&0&u\end{bmatrix}=S_1^{-1}\Lambda_A S_1$$

$$=\begin{bmatrix}1&\frac{\gamma}{2c^2}&0&0&\frac{\gamma}{2c^2}\\0&\frac{\gamma}{2\rho c}&0&0&-\frac{\gamma}{2\rho c}\\0&0&1&0&0\\0&0&0&1&0\\0&\frac{1}{2}&0&0&\frac{1}{2}\end{bmatrix}\begin{bmatrix}u&&&&\\&u+c&&&\\&&u&&\\&&&u&\\&&&&u-c\end{bmatrix}\begin{bmatrix}1&0&0&0&-\frac{\gamma}{c^2}\\0&\frac{\rho c}{\gamma}&0&0&1\\0&0&1&0&0\\0&0&0&1&0\\0&-\frac{\rho c}{\gamma}&0&0&1\end{bmatrix},$$

$$B_1=\begin{bmatrix}v&0&\rho&0&0\\0&v&0&0&0\\0&0&v&0&\frac{\gamma}{\rho}\\0&0&0&v&0\\0&0&p&0&v\end{bmatrix}=T_1^{-1}\Lambda_B T_1$$

$$=\begin{bmatrix}1&0&\frac{\gamma}{2c^2}&0&\frac{\gamma}{2c^2}\\0&1&0&0&0\\0&0&\frac{\gamma}{2\rho c}&0&\frac{\gamma}{2\rho c}\\0&0&0&1&0\\0&0&\frac{1}{2}&0&\frac{1}{2}\end{bmatrix}\begin{bmatrix}v&&&&\\&v+c&&&\\&&v&&\\&&&v&\\&&&&v-c\end{bmatrix}\begin{bmatrix}1&0&0&0&-\frac{\gamma}{c^2}\\0&1&0&0&0\\0&0&\frac{\rho c}{\gamma}&0&1\\0&0&0&1&0\\0&0&-\frac{\rho c}{\gamma}&0&1\end{bmatrix},\tag{1-19}$$

$$C_1=\begin{bmatrix}w&0&0&\rho&0\\0&w&0&0&0\\0&0&w&0&0\\0&0&0&w&\frac{\gamma}{\rho}\\0&0&0&p&w\end{bmatrix}=Q_1^{-1}\Lambda_C Q$$

$$
= \begin{bmatrix} 1 & 0 & 0 & \frac{\gamma}{2c^2} & \frac{1}{2c^2} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{\gamma}{2\rho c} & -\frac{\gamma}{2\rho c} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} w & & & & \\ & w+c & & & \\ & & w & & \\ & & & w & \\ & & & & w-c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & -\frac{\gamma}{c^2} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{\rho c}{\gamma} & 1 \\ 0 & 0 & 0 & -\frac{\rho c}{\gamma} & 1 \end{bmatrix}. \tag{1-20}
$$

1.2 特征理论

一般地，通过降阶的方法可以将高阶偏微分方程(组)改写为一阶偏微分方程(组). 所以不失一般性，下面只讨论一阶偏微分方程组.

与时间相关的一维问题的一阶偏微分方程组具有如下的形式:

$$
\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x} = \boldsymbol{R}, \tag{1-21}
$$

它的特征方程为

$$
\det | \boldsymbol{A} - \lambda \boldsymbol{I} | = 0, \quad \lambda = \frac{\mathrm{d}x}{\mathrm{d}t}. \tag{1-22}
$$

令用 $\boldsymbol{R} - \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t}$ 替代矩阵 $\boldsymbol{A} - \lambda \boldsymbol{I}$ 中任意一列所得矩阵的行列式的值为零，就得到特征值的关系式. 这是一个微分关系，它积分后得到的积分常数就是黎曼(Riemann)不变量.

一维不定常可压缩无粘性流体的欧拉(Euler)方程为

$$
\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ u \end{bmatrix} + \begin{bmatrix} u & 1 \\ \frac{a^2}{\rho} & u \end{bmatrix}\frac{\partial}{\partial x}\begin{bmatrix} \rho \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1-23}
$$

特征值为 $\lambda = u + a, u - a,$

特征值的关系式为

$$
\begin{vmatrix} u - \lambda & -\frac{\mathrm{d}\rho}{\mathrm{d}t} \\ \frac{a^2}{\rho} & -\frac{\mathrm{d}u}{\mathrm{d}t} \end{vmatrix} = 0
$$

或

$$
\frac{a\mathrm{d}\rho}{\rho} \pm \mathrm{d}u = 0.
$$

积分得

$$
\frac{2a}{\gamma - 1} \pm u = \text{const}.
$$

它就是黎曼不变量，分别对应特征值 $u \pm a$.

多维问题的偏微分方程组的一般形式为

$$\sum_{i=1}^{m} A_i \frac{\partial U}{\partial x_i} = R, \tag{1-24}$$

其中 $$U = (u_1, u_2, \cdots, u_n)^{\mathrm{T}}, \quad R = (r_1, r_2, \cdots, r_n)^{\mathrm{T}}, \tag{1-25}$$

这是 m 维 n 元的多维偏微分方程组.若作坐标旋转,即由 x 转到 z,二者的关系为

$$z_l = n_{l_1} x_1 + n_{l_2} x_2 + \cdots + n_{l_m} x_m = \sum_{k=1}^{m} n_{l_k} x_k, \tag{1-26}$$

$$l = 1, 2, \cdots, m,$$

其中 n_{l_k} 是 z_l 坐标方向与 x_k 坐标方向间的夹角的余弦.由上式可得

$$\frac{\partial}{\partial x_l} = \sum_{k=1}^{m} \frac{\partial}{\partial z_k} \frac{\partial z_k}{\partial x_l} = \sum_{k=1}^{m} n_{k_l} \frac{\partial}{\partial z_k}, \tag{1-27}$$

由此,(1-24) 式可改写为

$$\sum_{i=1}^{m} A_i n_{\nu_i} \frac{\partial U}{\partial z_\nu} + \sum_{i=1}^{m} \sum_{k=1}^{m}{}' A_i n_{k_i} \frac{\partial U}{\partial z_k} = R, \tag{1-28}$$

其中 ν 是 z 坐标系中某一 z_ν 方向. $\sum\limits_{k=1}^{m}{}'$ 表示 k 从 1 到 m 求和,但不包括 ν 方向.上式进一步可改写为

$$\sum_{i=1}^{m} A_i n_{\nu_i} \frac{\partial U}{\partial z_\nu} = R - \sum_{i=1}^{m} \sum_{k=1}^{m}{}' A_i n_{k_i} \frac{\partial U}{\partial z_k}. \tag{1-29}$$

当

$$\det \left| \sum_{i=1}^{m} A_i n_{\nu_i} \right| = 0 \tag{1-30}$$

时,ν 为特征面的法向.记特征向量为 $\boldsymbol{\lambda}$,则有

$$\boldsymbol{\lambda} = |\boldsymbol{\lambda}| \boldsymbol{n}_\nu,$$

这里 $\boldsymbol{n}_\nu$ 为归一化的单位向量.因 $\boldsymbol{\lambda} = (\lambda, \lambda_2, \cdots, \lambda_m)^{\mathrm{T}}$,故(1-30) 式可改写为

$$\det \left| \sum_{i=1}^{m} A_i \lambda_i \right| = 0, \tag{1-31}$$

这就是多维问题的特征方程.若记 $A_i = [a_{i;k,l}]_{n \times n}$ 则上式可改写为

$$\det | \boldsymbol{\lambda} \cdot \boldsymbol{a}_{k,l} |_{n \times n} = 0, \quad \boldsymbol{a}_{k,l} = (a_{1;k,l}, a_{2;k,l}, \cdots, a_{n;k,l})^{\mathrm{T}}. \tag{1-32}$$

特征值的关系,即特征相容条件,可以改述为(1-29) 式右端向量列与左端系数阵 $\sum\limits_{i=1}^{m} A_i n_{\nu_i}$ 一起构成的增广矩阵的行间应当是线性相关的.换言之,存在一个不全为零的数组 $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)^{\mathrm{T}}$,使得

$$\begin{cases} \alpha_1 \boldsymbol{\lambda} \cdot \boldsymbol{a}_{1,1} + \alpha_2 \boldsymbol{\lambda} \cdot \boldsymbol{a}_{2,1} + \cdots + \alpha_n \boldsymbol{\lambda} \cdot \boldsymbol{a}_{n,1} = 0, \\ \alpha_1 \boldsymbol{\lambda} \cdot \boldsymbol{a}_{1,2} + \alpha_2 \boldsymbol{\lambda} \cdot \boldsymbol{a}_{2,2} + \cdots + \alpha_n \boldsymbol{\lambda} \cdot \boldsymbol{a}_{n,2} = 0, \\ \cdots\cdots \\ \alpha_1 \boldsymbol{\lambda} \cdot \boldsymbol{a}_{1,n} + \alpha_2 \boldsymbol{\lambda} \cdot \boldsymbol{a}_{2,n} + \cdots + \alpha_n \boldsymbol{\lambda} \cdot \boldsymbol{a}_{n,n} = 0, \\ \boldsymbol{\alpha} \cdot \left(R - \sum\limits_{i=1}^{m} \sum\limits_{k=1}^{m}{}' A_i n_{k_i} \dfrac{\partial U}{\partial z_k} \right)^{\mathrm{T}} = 0, \end{cases} \tag{1-33}$$

或简写为

$$\begin{cases} \boldsymbol{\alpha}\cdot[\boldsymbol{\lambda}\cdot \boldsymbol{a}_{k,l}]^{\mathrm{T}}_{n\times n}=\boldsymbol{0}, \\ \boldsymbol{\alpha}\cdot\left(\boldsymbol{R}-\sum_{i=1}^{m}\sum_{k=1}^{m}{}' A_i n_{k_i}\dfrac{\partial \boldsymbol{U}}{\partial z_k}\right)^{\mathrm{T}}=0. \end{cases} \tag{1-34}$$

由于(1-34)式中第一式可以改写为 $\boldsymbol{\alpha}\cdot\left(\sum_{i=1}^{m}A_i n_{v_i}\right)^{\mathrm{T}}=0$,它与第二式相减得

$$\boldsymbol{\alpha}\cdot\left(\boldsymbol{R}-\sum_{i=1}^{m}\boldsymbol{A}_i\frac{\partial \boldsymbol{U}}{\partial x_i}\right)^{\mathrm{T}}=0, \tag{1-35}$$

这就是特征相容条件.一个特征向量 $\boldsymbol{\lambda}$ 对应一个 $\boldsymbol{\alpha}$. n 元问题有 n 个特征向量,应对应 n 个 $\boldsymbol{\alpha}$.

以二维不定常理想完全气体等熵流动的方程

$$\boldsymbol{A}_t\frac{\partial \boldsymbol{U}}{\partial t}+\boldsymbol{A}_x\frac{\partial \boldsymbol{U}}{\partial x}+\boldsymbol{A}_y\frac{\partial \boldsymbol{U}}{\partial y}=0$$

为例,其中 $\boldsymbol{U}=(\rho,u,v,p)^{\mathrm{T}}$,

$$\boldsymbol{A}_t=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\-a^2&0&0&1\end{bmatrix},\boldsymbol{A}_x=\begin{bmatrix}u&0&0&0\\0&u&0&\dfrac{1}{\rho}\\0&0&u&0\\-a^2u&0&0&u\end{bmatrix},\boldsymbol{A}_y=\begin{bmatrix}v&0&0&0\\0&v&0&0\\0&0&v&\dfrac{1}{\rho}\\-a^2v&0&0&v\end{bmatrix}.$$

特征方程为 $\det|\lambda_t\boldsymbol{A}_t+\lambda_x\boldsymbol{A}_x+\lambda_y\boldsymbol{A}_y|=0$,

展开得 $d^2[d^2-a^2(\lambda_x^2+\lambda_y^2)]=0$, $d=\lambda_t+u\lambda_x+v\lambda_y$,

方程的根为

$$\begin{cases} d_1=\lambda_t+u\lambda_x+v\lambda_y=0, \\ d_2=\lambda_t+u\lambda_x+v\lambda_y=0, \\ d_3=\lambda_t+u\lambda_x+v\lambda_y=a\sqrt{\lambda_x^2+\lambda_y^2}, \\ d_4=\lambda_t+u\lambda_x+v\lambda_y=-a\sqrt{\lambda_x^2+\lambda_y^2}. \end{cases}$$

由于特征值本身的大小并不重要,其方向是重要的,所以可以设 $\sqrt{\lambda_x^2+\lambda_y^2}=1$, $\lambda_x=\cos\theta$, $\lambda_y=\sin\theta$,于是四个特征值为

$$\begin{cases} \lambda_t+u\cos\theta+v\sin\theta=0, \\ \lambda_t+u\cos\theta+v\sin\theta=0, \\ \lambda_t+u\cos\theta+v\sin\theta=a, \\ \lambda_t+u\cos\theta+v\sin\theta=-a. \end{cases}$$

为确定 $\boldsymbol{\alpha}$,要找出使

$$\lambda_t\boldsymbol{A}_t+\lambda_x\boldsymbol{A}_x+\lambda_y\boldsymbol{A}_y=\begin{bmatrix}d&\rho\lambda_x&\rho\lambda_y&0\\0&d&0&\lambda_x/\rho\\0&0&d&\lambda_y/\rho\\-a^2d&0&0&d\end{bmatrix}$$

行间线性相关的系数. 不难验证下列 $\boldsymbol{\alpha}$ 与对应 $\boldsymbol{\lambda}$ 使相应条件满足

$$\begin{cases} \boldsymbol{\alpha}^{(1)} = (0,\lambda_y,-\lambda_x,0) = (0,\sin\theta,-\cos\theta,0), \\ \boldsymbol{\alpha}^{(2)} = (0,0,0,1), \\ \boldsymbol{\alpha}^{(3)} = (a^2,-\rho a\lambda_x,-\rho a\lambda_y,1) = (a^2,-\rho a\cos\theta,-\rho a\sin\theta,1), \\ \boldsymbol{\alpha}^{(4)} = (a^2,\rho a\lambda_x,\rho a\lambda_y,1) = (a^2,\rho a\cos\theta,\rho a\sin\theta,1). \end{cases}$$

相应的特征相容条件为

$\boldsymbol{\alpha}^{(1)}$ 对应 $$\sin\theta\left(\frac{\mathrm{d}u}{\mathrm{d}t}+\frac{1}{\rho}\frac{\partial p}{\partial x}\right)-\cos\theta\left(\frac{\mathrm{d}v}{\mathrm{d}t}+\frac{1}{\rho}\frac{\partial p}{\partial y}\right)=0;$$

$\boldsymbol{\alpha}^{(2)}$ 对应 $$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{p}{\rho^\gamma}\right)=0;$$

$\boldsymbol{\alpha}^{(3)}$ 对应 $$a^2\left[\frac{\mathrm{d}\rho}{\mathrm{d}t}+\rho(\nabla\cdot\boldsymbol{v})\right]-\rho a\cos\theta\left(\frac{\mathrm{d}u}{\mathrm{d}t}+\frac{1}{\rho}\frac{\partial p}{\partial x}\right)-\rho a\sin\theta\left(\frac{\mathrm{d}v}{\mathrm{d}t}+\frac{1}{\rho}\frac{\partial p}{\partial y}\right)+\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{p}{\rho^\gamma}\right)=0;$$

$\boldsymbol{\alpha}^{(4)}$ 对应 $$a^2\left[\frac{\mathrm{d}\rho}{\mathrm{d}t}+\rho(\nabla\cdot\boldsymbol{v})\right]+\rho a\cos\theta\left(\frac{\mathrm{d}u}{\mathrm{d}t}+\frac{1}{\rho}\frac{\partial p}{\partial x}\right)+\rho a\sin\theta\left(\frac{\mathrm{d}v}{\mathrm{d}t}+\frac{1}{\rho}\frac{\partial p}{\partial y}\right)+\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{p}{\rho^\gamma}\right)=0.$$

1.3 间断条件

在双曲型方程为线性时,则特征值都是常数,所以同族特征是互相平行的直线或锥面,它们是不会相交的. 但是在非线性问题中,特征都是曲线或形状复杂的曲面,它们之间有可能相交. 另外这时特征的几何位置要与方程的解同时确定,所以事先不知其形状,也不知是否会相交以及相交的位置. 由于同族特征满足相同的相应的特征关系,在相交时要求不同值满足同一条件,这显然是不可能的. 这就意味着解可能出现不连续,即间断. 当然间断位置不与同族特征相交的位置一致,它的位置也由求解过程同时确定.

在有间断的地方,微分方程就不再成立,但是积分型的方程仍然满足. 所以可以借用积分型方程得到间断前后物理量间的关系式. 对于一维情况,方程组可以写作

$$\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}=\boldsymbol{0}, \tag{1-36}$$

相应的间断条件为

$$[\boldsymbol{U}]D=\boldsymbol{F},\quad D=\frac{\mathrm{d}x}{\mathrm{d}t}. \tag{1-37}$$

对于气体动力学方程组,

$$\boldsymbol{U}=(\rho,\rho u,\rho E_s)^{\mathrm{T}},\quad \boldsymbol{F}=(\rho u,\rho u^2+p,u(\rho E_s+p))^{\mathrm{T}},$$

于是相应的间断条件变为

$$\begin{cases}[\rho]D = [\rho u],\\ [\rho u]D = [\rho u^2 + p],\\ [\rho E_s]D = [\rho E_s u + pu].\end{cases} \tag{1-38}$$

若记 $m = \rho(D - u)$，则有

$$[m] = 0,\quad m[u] = [p],\quad m[E_s] = [pu]. \tag{1-39}$$

可见 $[u] = 0$，$[p] = 0$ 时有 $u_1 = u_2$，$p_1 = p_2$，自动得 $[E_s] = 0$ 或 $E_{s_1} = E_{s_2}$. 这里对 ρ 和 T 值无约束，在间断两侧可以相等，也可以不相等. 当两侧值均相等时就不成其为间断，属无聊解. 当左右两侧速度、压力相等而密度、温度不等时，称作**接触间断**.

更为一般的情况是各个物理量均有间断，叫作**强间断**. 在可压缩流体流动中这种间断就是**激波**. 由(1-39)式可知流体穿过激波时的流量是不变的. 引入记号 $v = D - u$，则激波前后的物理量的关系式为

$$\begin{cases}\rho_1 v_1 = \rho_2 v_2,\\ \rho_1 v_1^2 + p_1 = \rho_2 v_2^2 + p_2,\\ i_1 + \dfrac{1}{2}v_1^2 = i_2 + \dfrac{1}{2}v_2^2,\end{cases} \tag{1-40}$$

其中 $i = e + p/\rho$ 为焓，$e = C_V T$ 为内能. 由上关系可以得到如下的关系：

$$\frac{\rho_1}{\rho_2} = \frac{(\gamma+1)p_1 + (\gamma-1)p_2}{(\gamma-1)p_1 + (\gamma+1)p_2} \Leftrightarrow \frac{p_2 - p_1}{\rho_2 - \rho_1} = \gamma\frac{p_2 + p_1}{\rho_2 + \rho_1} \tag{1-41}$$

$$\rho v = \rho_1 a_1\sqrt{\frac{\gamma+1}{2\gamma}\frac{p_2}{p_1} + \frac{\gamma-1}{2\gamma}} = \rho_2 a_2\sqrt{\frac{\gamma+1}{2\gamma}\frac{p_1}{p_2} + \frac{\gamma-1}{2\gamma}}, \tag{1-42}$$

$$v_1 v_2 = (a^*)^2,$$

$$D = u_1 + a_1\sqrt{\frac{\gamma-1}{2\gamma} + \frac{\gamma+1}{2}\frac{p_2}{p_1}} = u_2 + a_2\sqrt{\frac{\gamma-1}{2\gamma} + \frac{\gamma+1}{2}\frac{p_1}{p_2}}. \tag{1-43}$$

(1-41)式又称为 **Hugoniot 关系**. 由(1-41)—(1-43)式可以看出**间断两侧具有等价性**，即除了无聊解以外还存在两个解，即 $p_2 > p_1$ 和 $p_2 < p_1$. 但考虑到流动是绝热的，流体通过激波时熵只能增加不能减少. 设 1 表示激波前的状态，即流体尚未通过激波的状态，2 表示激波后的状态. 则由熵增原理可知必须有 $p_2 > p_1$，$\rho_2 > \rho_1$，$v_2 < v_1$，$M_2 = \dfrac{v_2}{a_2} < 1 < M_2 = \dfrac{v_1}{a_1}$. 这就是气体运动中确定激波的**熵条件**.

1.4 黎曼问题的解

有两个不同的各自均匀的流场，在 $t = 0$ 时刻在某处接触构成间断(该处可设为 $x = 0$). 以该有间断的流场作为初始场的一维不定常气体动力学方程(1-36)求解问题，称作**黎曼问题**. 可表述如下：

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = \boldsymbol{0}, \tag{1-44}$$

$$\boldsymbol{U}(x,0) = \begin{cases}\boldsymbol{U}_L, & x < 0,\\ \boldsymbol{U}_R, & x > 0,\end{cases} \tag{1-45}$$

其中 $U=(\rho,u,p)^{\mathrm{T}}$，$F=(\rho u,\rho u^2+p,(\rho E_s+p)u)^{\mathrm{T}}$. 该黎曼问题可有五种不同的类型，如图 1-1 所示.

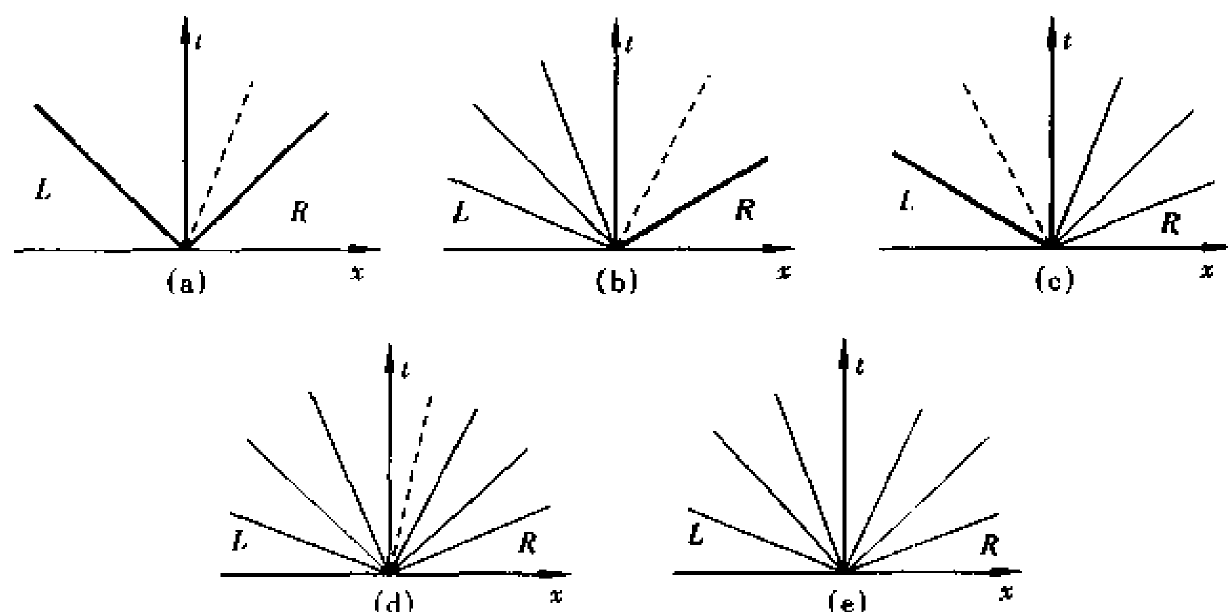


图 1-1

其中虚线为接触间断，细实线为膨胀波，粗线为激波. 由图不难看出这五种情况都是由左传播波和右传播波组成的. 设接触间断面上的速度为 U，压力为 P，则

$$u_L=\begin{cases} U-\dfrac{c_L}{\gamma}\left[1-\left(\dfrac{P}{p_L}\right)^{\frac{\gamma-1}{2\gamma}}\right], & P<p_L,\\[2ex] U-\dfrac{P-p_L}{\rho_L c_L\sqrt{\dfrac{\gamma+1}{2\gamma}\left(\dfrac{P}{p_L}-1\right)+1}}, & P>p_L,\end{cases}$$

$$u_R=\begin{cases} U+\dfrac{c_R}{\gamma}\left[1-\left(\dfrac{P}{p_R}\right)^{\frac{\gamma-1}{2\gamma}}\right], & P<p_R,\\[2ex] U+\dfrac{P-p_R}{\rho_R c_R\sqrt{\dfrac{\gamma+1}{2\gamma}\left(\dfrac{P}{p_R}-1\right)+1}}, & P>p_R.\end{cases}$$

U、P 可由图 1-2 得到.

这五种情况分别对应

(1) 分解成左右传播的两个行进激波. 它对应于两个激波对撞后的情况.

(2) 分解成右传播激波和左传播膨胀波.

(3) 分解成左传播激波和右传播膨胀波.

(4) 分解成左右传播的膨胀波，对应流体向两侧外流.

(5) 同(4) 情况，但中间产生真空区.

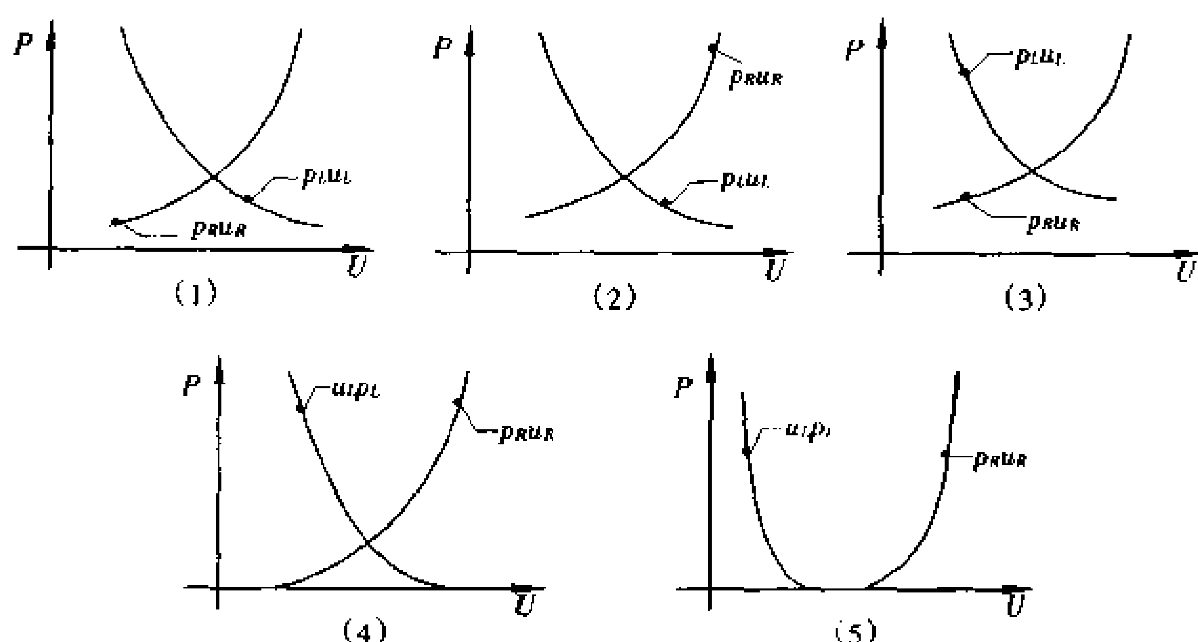


图 1-2

2 发展方程的有限差分法

描写随时间变化的物理过程的数学方程称为发展方程. 如热传导方程、热扩散方程、不定常纳维(Navier)(Stokes)-斯托克斯方程、不定常欧拉方程和波动方程等. 一般地说,发展方程可以分成两大类:与扩散相关的方程,一般具有抛物型特点;与波动相关而不计其耗散的方程,一般具有双曲型特点. 对于复杂问题往往两者兼而有之.

有限差分法是用差商代替微商,从而达到方程离散化的目的. 通过求解离散化的代数方程组得到离散点上的解的近似值. 可见所得的解只在离散点近似地满足方程. 这种离散的代数方程叫作有限差分方程.

2.1 差分格式的适定性,拉克斯定理

一般地讲,边界点建立差分方程的方法和内点建立差分方程的方法是不一致的. 在求解微分方程时,将建立的内点和边界点的差分方程叫作**差分格式**. 但实际上,人们一般将内点差分方程叫作差分格式. 所以以下也将采用这种简单的称呼.

以热传导方程的初边值问题为例,方程及其边界条件为

$$\begin{cases}\dfrac{\partial u}{\partial t} = \nu\dfrac{\partial^2 u}{\partial x^2}, & t > 0, 0 \leqslant x \leqslant L,\\ u(x,0) = f(x), \\ u(0,t) = \varphi_1(t), \quad u(L,t) = \varphi_2(t).\end{cases} \tag{2-1}$$

为确定起见，求解范围限于 $0 \leqslant t \leqslant T$，$T$ 为某一确定时刻. 用差分法求解时，首先要将求解域用网格进行划分. 设在 $(0, L)$ 之间划分为 J 等分，$(0, T)$ 之间划分成 N 等分，如图 2-1 所示. 于是有

$$\Delta x = L/J, \quad \Delta t = T/N,$$
$$x_j = j\Delta x, \quad t_n = n\Delta t,$$
$$j = 0, 1, \cdots, J;\ n = 0, 1, \cdots, N.$$

(j, n) 点上的值记作 u_j^n，该点的微商用差商近似时，可有不同的计算差商的方法，不同的近似方法就形成了不同的差分格式. 最简单的办法是

$$\left(\frac{\partial u}{\partial t}\right)_j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t},$$
$$\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}. \tag{2-2}$$

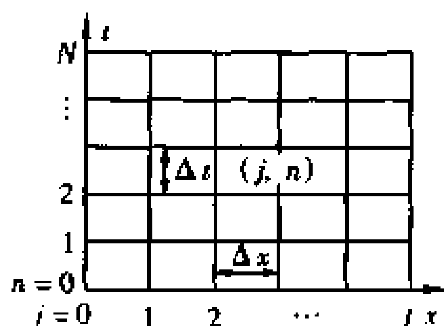


图 2-1

将它们代入(2-1) 式后，可得差分格式如下：

$$\begin{cases} \dfrac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \dfrac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}, \\ u_j^0 = u(x_j, 0), \\ u_0^n = u(0, n\Delta t) = \varphi_1(n\Delta t), \\ u_J^n = u(L, n\Delta t) = \varphi_2(n\Delta t). \end{cases} \tag{2-3}$$

(2-3) 式中的第一式可改写为

$$\begin{cases} u_j^{n+1} = \sigma u_{j+1}^n + (1 - 2\sigma) u_j^n + \sigma u_{j-1}^n, \quad n \geqslant 0, j = 1, 2, \cdots, J - 1, \\ \sigma = \dfrac{\nu \Delta t}{\Delta x}. \end{cases} \tag{2-4}$$

在这里时间差分用的是向前差分，空间差分用的是中心差分，所以该格式又叫做 FTCS(forward in time, center in space) 格式. 一般地，差分格式可以表示为

$$\mathrm{L}_h u_j^n = 0, \tag{2-5}$$

其中 L_h 表示离散化后的算子. 原来的微分方程记作

$$\mathrm{L}u = 0. \tag{2-6}$$

设 u 为微分方程的解，u_j^n 是差分方程在 (x_j, t_n) 处的解，记 $(ET)_j^n = \mathrm{L}_h u_j^n - (\mathrm{L}u)_j^n$ 为差分格式的截断误差. 若对于一个足够光滑的函数 u，当 $\Delta x \to 0$、$\Delta t \to 0$ 时的截断误差对于每一个 (j, n) 点都趋于零时，则称差分格式 $\mathrm{L}_h u_j^n = 0$ 逼近微分方程 $\mathrm{L}u = 0$，即差分格式与原微分方程相容.

利用泰勒(Taylor) 展开，不难得到(2-3) 式中第一式为

$$\left(\frac{\partial u}{\partial t}\right)_j^n = \nu\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n + O(\Delta t) + O(\Delta x^2), \tag{2-7}$$

所以 $(ET)_j^n = O(\Delta t) + O(\Delta x^2)$. 当 Δt、$\Delta x \to 0$ 时，$(ET)_j^n \to 0\ (\forall j, n)$，所以 FTCS 格式与原微分方程是相容的. 截断误差的时间方向为一阶，空间方向为二阶.

差分格式与微分方程相容是保证差分方程所得的近似解在离散点上逼近真解的必要条件. 事实上，为使差分方程能得到逼近真解的近似解，还必须要求差分格

式是收敛的、稳定的及解对边界条件有连续依赖的关系.

记差分格式(2-5)的解和微分方程(2-6)的解之间的差为 $e_j^n = u_j^n - u(x_j, t_n)$，若对于相当广泛一类定解函数(初值和边值函数)，下式都成立

$$u_j^n - u(x_j, t_n) \xrightarrow[\forall j, n]{\Delta x, \Delta t \to 0} 0, \tag{2-8}$$

则称该差分格式的解收敛.

设在 t_0 时刻差分格式的解为 u_j^0，但在实际计算中由于某种原因(如计算机舍入误差，数据误差，小扰动等)得到的值为 $(u)_j^0$，二者之间的差为 $\delta u_j^0 = |(u)_j^0 - u_j^0|$，在 t_n 时刻 $\delta u_j^n = |(u)_j^n - u_j^n|$. 如果存在一个与 n 无关的有界值 K，使得

$$\sup_{\forall j} |\delta u_j^n| \leqslant K \sup_{\forall j} |\delta u_j^0|,$$

则称**差分格式稳定**.

若差分格式与微分方程是相容的，差分方程本身是收敛的、稳定的，及差分方程的解对定解条件有连续依赖关系，则称**差分格式适定**.

对于线性微分方程，有如下的拉克斯(Lax)定理：

定理 1　若线性微分方程的初值问题是适定的，相应的差分格式是相容的，则差分格式的稳定性与收敛性是等价的，即

$$\text{收敛性} \xleftrightarrow[\text{差分方程与微分方程相容}]{\text{线性方程、适定的初值问题}} \text{稳定性}$$

微分方程的适定性是指微分方程在给定的定解条件下存在唯一解，并对定解条件连续依赖.

证明　记发展方程为 $\frac{\partial u}{\partial t} = \mathrm{L}u$，相应的差分格式为 $u^{n+1} = \mathrm{L}_h u^n$，并有 $u^{n+1} = \mathrm{E}(\Delta t)u^n + O(\Delta t^{\nu_1+1}) + O(\Delta x^{\nu_2+1})$，其中 ν_1、ν_2 为正数，$\mathrm{E}(\Delta t)$ 为时间方向算子，有如下性质：

$$\mathrm{E}(0) = I, \quad \mathrm{E}(t_1 + t_2) = \mathrm{E}(t_1)\mathrm{E}(t_2).$$

由稳定性定义知，设本格式是稳定的，则在 $0 \leqslant n\Delta t \leqslant T$ 范围内有

$$\| \mathrm{L}_h^{(n)} u_j^0 \| < K \| u_j^0 \|,$$

其中 $\| \cdot \|$ 为范数，K 是与 j、n 无关的有限数. 又由适定性知，若初值为 f，则

$$\| \mathrm{E}(t) f \| < K \| f \|.$$

收敛性要求

$$\| u_j^n - u(x_j, t_n) \| = \| [\mathrm{L}_h^{(n)} - \mathrm{E}(t)] f \| \xrightarrow{\Delta t \to 0} 0.$$

(1) 证收敛 → 稳定

由收敛性定义可知

$$\| \mathrm{L}_h^{(n)} f \| < K \| f \|.$$

由于 $u_j^0 = f$，故 $\| \mathrm{L}_h^{(n)} u_j^0 \| < K \| u_j^0 \|$，即稳定性条件得以满足.

(2) 证稳定 → 收敛

由差分格式相容性知 $\Delta x / \Delta t < \alpha$，$\Delta t \to 0$ 时 $\| [\mathrm{L}_h - \mathrm{E}(t)] u \| \to 0$,

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} \| [\mathrm{L}_h - \mathrm{E}(\Delta t)] u \| = 0 \qquad (\text{注意 } \nu_1 > 0).$$

或 引入记号$[\mathrm{L}_h^{(n)} - \mathrm{E}(n\Delta t)]f = \psi_{n\Delta t}$,有

$$\begin{aligned}
\psi_{n\Delta t} &= [\mathrm{L}_h^{(n)} - \mathrm{L}_h^{(n-1)}\mathrm{E}(\Delta t) + \mathrm{L}_h^{(n-1)}\mathrm{E}(\Delta t) - \mathrm{L}_h^{(n-2)}\mathrm{E}(2\Delta t) + \cdots + \\
&\quad \mathrm{L}_h\mathrm{E}((n-1)\Delta t) - \mathrm{E}(n\Delta t)]f \\
&= [\mathrm{L}_h^{(n-1)}[\mathrm{L}_h - \mathrm{E}(\Delta t)] + \mathrm{L}_h^{(n-2)}[\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}(\Delta t) + \cdots + \\
&\quad [\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1)\Delta t)]f \\
&= \sum_{m=0}^{n-1} \mathrm{L}_h^{(m)}[\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1-m)\Delta t)f.
\end{aligned}$$

由稳定性条件知,

$$\begin{aligned}
\| \psi_{n\Delta t} \| &= \| \sum_{m=0}^{n-1} \mathrm{L}_h^{(m)}[\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1-m)\Delta t)f \| \\
&\leqslant K \| \sum_{m=0}^{n-1} [\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1-m)\Delta t)f \| \\
&\leqslant K \sum_{m=0}^{n-1} \| [\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1-m)\Delta t)f \| \\
&= K\Delta t \sum_{m=0}^{n-1} \frac{1}{\Delta t} \| [\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1-m)\Delta t)f \| \\
&= Kn\Delta t \cdot \max_{\forall m} \left| \frac{1}{\Delta t} \| [\mathrm{L}_h - \mathrm{E}(\Delta t)]\mathrm{E}((n-1-m)\Delta t)f \| \xrightarrow{\Delta t \to 0} 0, \right.
\end{aligned}$$

$$\lim_{\Delta t \to 0} \| [\mathrm{L}_h^{(n)} - \mathrm{E}(n\Delta t)]f \| = 0,$$

即

$$\| u_j^n - u(x_j, t) \| \xrightarrow{\Delta t \to 0} 0.$$

也就是说格式收敛、证毕.

由上分析可知:对于线性问题,收敛性与稳定性是等价的,只需分析其中之一即可,习惯上人们首先分析稳定性.另外注意到拉克斯定理只适用于线性问题,对于非线性问题,目前尚无相应的定理,所以只能作局部线化而得到一些有启发性的分析结果,最后验证收敛性和稳定性,由实际计算得到.

2.2 差分格式的稳定性分析

下列各种分析方法都只适用于线性问题,对于非线性问题则采用局部线化进行分析,所得结果作为非线性问题的有启发性的分析.

2.2.1 矩阵分析法

为方便起见,差分格式写作

$$u^{n+1} = E_h u^n. \tag{2-9}$$

设 $t=0$ 时的近似解 $\tilde{u}^0 = u^0 + \varepsilon^0$,其中 ε^0 为 $t=0$ 时的误差.假定在以后的计算

中不再引入新的误差,则有

$$\tilde{u}^{n+1} = E_h \tilde{u}^n. \tag{2-10}$$

(2-10) 式减去(2-9) 式得

$$\varepsilon^{n+1} = E_h \varepsilon^n,$$

或进而得

$$\varepsilon^{n+1} = E_h^{n+1} \varepsilon^0.$$

保证稳定的条件是 $\| E_h \| \leqslant 1$, E_h 的模可取为$\max\limits_{\forall i} |\lambda_i|$,其中 λ_i 为矩阵E_h 的特征值. 所以稳定条件为

$$\sup_{\forall j,n}\{\max_{\forall i} |\lambda_i|\} \leqslant 1. \tag{2-11}$$

以格式(2-4) 为例, E_h 为

$$E_h = \begin{bmatrix} 1-2\sigma & \sigma & & & \\ \sigma & 1-2\sigma & \sigma & & \\ & & \ddots & & \\ & & \sigma & 1-2\sigma & \sigma \\ & & & \sigma & 1-2\sigma \end{bmatrix},$$

它的特征值为

$$\lambda_i = 1 - 2\sigma + 2\sigma\cos\frac{(i-1)\pi}{n} = 1 - 4\sigma\sin^2\left[\frac{(i-1)\pi}{n}\right],$$

为使 λ_i 绝对值均不大于 1,则要求

$$|1 - 4\sigma| \leqslant 1$$

或

$$\sigma = \frac{\nu\Delta t}{\Delta x^2} \leqslant \frac{1}{2}. \tag{2-12}$$

这就是 FTCS 格式的稳定条件. 本方法由于确定矩阵特征值比较困难而在应用中受到限制.

2.2.2　冯·诺伊曼方法

首先设想将解作周期延拓,于是解可以用傅里叶级数表示出来,同样误差也可以用傅里叶级数进行展开. 如果该误差的每一个傅里叶分量在差分算子 E_h 的作用下都是衰减的,则整个误差也必将是衰减的,于是格式就是稳定的. 所以实际分析中只要对某一分量进行分析就可以了. 比如设

$$\varepsilon_j^n \sim e^{ikx_j}, \tag{2-13}$$

于是

$$\varepsilon_j^{n+1} \sim E_h e^{ikx_j} = G e^{ikx_j}, \tag{2-14}$$

其中 G 为放大因子. 当 G 的模对于所有的 k、j 都小于等于 1 时,格式就是稳定的.

事实上,设差分算子 E_h 表示为

$$u^{n+1} = E_h u_j^n = \sum_{l=-l_1}^{l_2} a_l u_{j+l}^n, \qquad l_1, l_2 \geqslant 0, \tag{2-15}$$

类似地有误差满足

$$\epsilon^{n+1} = E_h \epsilon_j^n = \sum_{l=-l_1}^{l_2} a_l \epsilon_{j+l}^n, \qquad l_1, l_2 \geqslant 0. \tag{2-16}$$

设
$$\epsilon^n(x) = \sum_{k=-\infty}^{+\infty} C_k e^{ikx}, \tag{2-17}$$

代入得
$$\epsilon^{n+1} = \sum_{k=-\infty}^{\infty} C_k \sum_{l=-l_1}^{l_2} a_l e^{ikx_{j+l}} = \sum_{-\infty}^{\infty} C_k G(k,j,\Delta x) e^{ikx_j}. \tag{2-18}$$

由于$\frac{1}{2\pi}\int_0^{2\pi} |\epsilon^{(n+1)(x)}|^2 dx = \sum_{-\infty}^{\infty} |C_k G(k,j,\Delta x)|^2$，当 (2-19)

$$\max_{\forall k,j} |G(k,j,\Delta t)| \leqslant 1 \tag{2-20}$$

时有

$$\frac{1}{2\pi}\int_0^{2\pi} |\epsilon^{(n+1)(x)}|^2 dx = \sum_{-\infty}^{\infty} |C_k G(k,j,\Delta x)|^2 \leqslant \sum_{-\infty}^{\infty} |C_k|^2 = \frac{1}{2\pi}\int_0^{2\pi} |\epsilon^{(n)}(x)|^2 dx,$$

即
$$\|\epsilon^{n+1}\| \leqslant \|\epsilon^n\|.$$

所以(2-13)式就是稳定条件.

用这一方法对格式(2-4)进行分析，记$\epsilon_j^n \sim e^{ikx_j}$，于是

$$\begin{aligned}
\epsilon_j^{n+1} &\sim \sigma e^{ikx_{j+1}} + (1-2\sigma)e^{ikx_j} + \sigma e^{ikx_{j-1}} \\
&= e^{ikx_j}[\sigma e^{ik\Delta x} + (1-2\sigma) + \sigma e^{-ik\Delta x}] \\
&= e^{ikx_j}[1 + 2(\cos k\Delta x - 1)\sigma] \\
&= e^{ikx_j}[1 - 4\sigma \sin^2(\frac{k\Delta x}{2})],
\end{aligned}$$

即
$$G = 1 - 4\sigma\sin^2(\frac{k\Delta x}{2}).$$

使所有的$|G|$都小于1的条件是

$$|1-4\sigma| \leqslant 1 \quad 或 \quad \sigma \leqslant \frac{1}{2}.$$

所得结论同矩阵分析法.不难看出本方法非常简便而实用.

2.2.3 Hirt 稳定性分析方法

该方法是一种启发性的分析方法，其基本思想是将差分方程在(j,n)点作泰勒展开，得到一双曲型近似方程，确定该点的依赖域.当近似方程的依赖域大于差分计算的依赖域时，格式才是稳定的.如格式(2-4)中各点值都在(j,n)处展开，得到

$$\begin{aligned}
&u_j^n + \left(\frac{\partial u}{\partial t}\right)_j^n \Delta t + \frac{1}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_j^n \Delta t^2 + \cdots \\
&= \sigma[u_j^n + \left(\frac{\partial u}{\partial x}\right)_j^n \Delta x + \frac{1}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n \Delta x^2 + \cdots] + (1-2\sigma)u_j^n + \\
&\quad \sigma[u_j^n - \left(\frac{\partial u}{\partial x}\right)_j^n \Delta x + \frac{1}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n \Delta x + \cdots]
\end{aligned} \tag{2-21}$$

或改写为

$$\left(\frac{\partial u}{\partial t}\right)_j^n = \frac{\sigma\Delta x^2}{\Delta t}\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n - \frac{\Delta t}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_j^n + \cdots$$
$$= \nu\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n - \frac{\Delta t}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_j^n + \cdots.$$

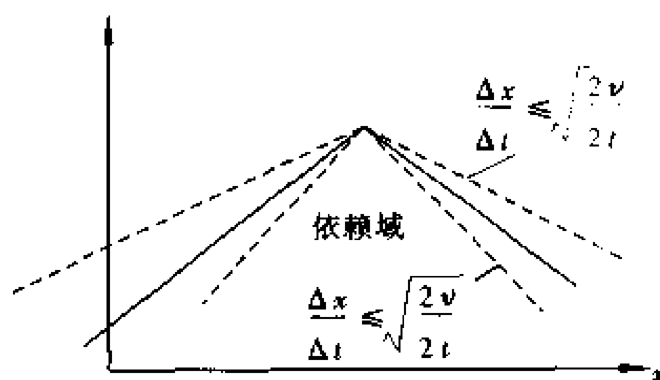


图 2-2

略去高阶项后得到一个双曲型方程，其特征线方程为$\frac{dx}{dt} = \pm\sqrt{\frac{2\nu}{\Delta t}}$，积分得特征线为$x = \pm t\sqrt{\frac{2\nu}{\Delta t}} + \text{const}$. 图 2-2 中虚线表示其特征线所构成的近似方程依赖域，实线为计算差分方程时的依赖域. 不难看出当

$$\frac{\Delta x}{\Delta t} \geqslant \sqrt{\frac{2\nu}{\Delta t}} \Rightarrow \sigma = \frac{\nu\Delta t}{\Delta x^2} \leqslant \frac{1}{2}$$

时才是稳定的.

2.2.4 修正方程判别法

本方法同样对差分方程中各项在(j,n)处作泰勒展开，但对关于t的高阶导数用关于x的各项导数加以表达，这一点用对所得近似方程反复求导可以得到. 如对格式(2-4) 中各项在(j,n)处作泰勒展开后得(2-21) 式，它可以简写为

$$\mathrm{L}_t u = \mathrm{L}_x u, \tag{2-22}$$

其中略去了标号j、n. 由于L_t、L_x均为线性算子，所以有

$$\mathrm{L}_t^l u = \mathrm{L}_x^l u, \tag{2-23}$$

其中l表示算子作用次数. 于是有

$$l = 4 \quad \left(\frac{\partial^4}{\partial t^4} + \cdots\right)u = \left(\nu^4\frac{\partial^8}{\partial x^8} + \cdots\right)u,$$

$$l = 3 \quad \left(\frac{\partial^3}{\partial t^3} + \frac{3}{2}\Delta t\frac{\partial^4}{\partial t^4} + \cdots\right)u = \nu^3\left(\frac{\partial^6}{\partial x^6} + \frac{\Delta x^2}{4}\frac{\partial^8}{\partial x^8} + \cdots\right)u,$$

$$l = 2 \quad \left(\frac{\partial^2}{\partial t^2} + \Delta t\frac{\partial^3}{\partial t^3} + \frac{7}{12}\Delta t^2\frac{\partial^4}{\partial t^4} + \cdots\right)u = \nu^2\left(\frac{\partial^4}{\partial x^4} + \frac{\Delta x^2}{6}\frac{\partial^6}{\partial t^6} + \frac{\Delta x^4}{80}\frac{\partial^8}{\partial x^8} + \cdots\right)u,$$

$$l = 1 \quad \left(\frac{\partial}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2}{\partial t^2} + \frac{\Delta t^2}{6}\frac{\partial^3}{\partial t^3} + \frac{\Delta x^4}{24}\frac{\partial^4}{\partial t^4} + \cdots\right)u$$

$$= \nu\left(\frac{\partial^2}{\partial x^2} + \frac{\Delta x^2}{12}\frac{\partial^4}{\partial x^4} + \frac{\Delta x^4}{360}\frac{\partial^6}{\partial x^6} + \frac{\Delta x^6}{20160}\frac{\partial^8}{\partial x^8} + \cdots\right)u.$$

用 $l = 2,3,4,\cdots$ 消去 $l = 1$ 中的关于 t 的二、三、四等阶导数，可得

$$\frac{\partial u}{\partial t} = (\nu + \nu_2)\frac{\partial^2 u}{\partial x^2} + \sum_{k=1}^{\infty}\nu_{2k+1}\frac{\partial^{2k+1}u}{\partial x^{2k+1}} + \sum_{k=2}^{\infty}\nu_{2k}\frac{\partial^{2k}u}{\partial x^{2k}}. \tag{2-24}$$

若设 $u = e^{\beta t}e^{i\alpha x}$，代入方程后得

$$\beta = -\alpha^2(\nu + \nu_2) + \sum_{k=1}^{\infty}(-1)^k i\alpha^{2k+1}\nu_{2k+1} + \sum_{k=2}^{\infty}(-1)^k\alpha^{2k}\nu_{2k}.$$

设 α、β 都是实数，故可以看出 ν_2、ν_{2k} 引起波的耗散，而 ν_{2k+1} 引起色散. 故为使耗散总为正，要求

$$\nu + \nu_2 > 0,\quad (-1)^k\nu_{2k} < 0,\quad k = 2,3,\cdots.$$

在上述例子中 $\nu_2 = 0$，$\nu_4 = \nu\left(\frac{\Delta x^2}{12} - \frac{\Delta t}{2}\nu\right)$，… 稳定条件为

$$\nu_4 > 0 \Rightarrow \frac{\Delta x^2}{12} - \frac{\Delta t}{2}\nu > 0 \Rightarrow \frac{\nu\Delta t}{\Delta x^2} < \frac{1}{6}.$$

这里得到的结果偏严格.

由上分析知(2-24)式与差分方程是等价的，与原求解的方程有差别，故称作**修正方程**. 用修正方程的方法作稳定性分析得到的将是充分条件.

2.3　一些常用的差分格式

对流扩散方程是最具代表性的发展方程，它具有如下形式：

$$\begin{cases}\dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = \nu\dfrac{\partial^2 u}{\partial x^2},\\ u(x,0) = \varphi(x),\\ u(0,t) = f_L(t), u(L,t) = f_R(x)\quad(\lambda \subset (0,L)).\end{cases} \tag{2-25}$$

其中设 $a > 0$，$\nu > 0$. 当 $\nu = 0$ 时为对流方程，也是双曲型方程；当 $a = 0$ 时为扩散方程.

2.3.1　扩散方程($a = 0$)

1. FTCS 格式(即(2-4)式)

$$u_j^{n+1} = \sigma u_{j+1}^n + (1 - 2\sigma)u_j^n + \sigma u_{j-1}^n. \tag{2-26}$$

稳定条件为

$$\sigma = \frac{\nu\Delta t}{\Delta x^2} \leqslant \frac{1}{2}.$$

2. BTCS 格式(即时间后向空间中心格式)

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2},$$

即

$$\sigma u_{j+1}^{n+1} - (1 + 2\sigma)u_j^{n+1} + \sigma u_{j-1}^{n+1} = u_j^n. \tag{2-27}$$

FTCS格式由第 n 层的值直接推进计算得到第 $n+1$ 层的解，故称显式格式. BTCS格式的第 $n+1$ 层的值需解联列方程得到，所以称为隐式格式. 利用冯·诺伊曼方法可以得到放大因子

$$G(k,\Delta x) = -\frac{1}{1+4\sigma\sin^2\left(\frac{k\Delta x}{2}\right)}. \tag{2-28}$$

不难看出 $|G|\leqslant 1$，所以格式无条件稳定.

3. Crank-Nicolson 格式

将以上两个格式加以组合，得

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} = \nu\left[\theta\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{\Delta x^2}+(1-\theta)\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{\Delta x^2}\right], \tag{2-29}$$

其中 $0\leqslant\theta\leqslant 1$. 用冯·诺伊曼方法分析得

$$G(k,\Delta x) = \frac{1-4(1-\theta)\sigma\sin^2\left(\frac{k\Delta x}{2}\right)}{1+4\theta\sigma\sin^2\left(\frac{k\Delta x}{2}\right)}, \tag{2-30}$$

稳定条件为

$$\begin{cases}\theta > \dfrac{1}{2}, & \text{无条件稳定},\\ \theta < \dfrac{1}{2},\quad 0<\sigma\leqslant\dfrac{1}{2-4\theta}, & \text{稳定}.\end{cases} \tag{2-31}$$

利用泰勒公式可得

$$\begin{aligned}&u_j^{n+1}-u_j^n-\sigma[\theta(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})+(1-\theta)(u_{j+1}^n-2u_j^n+u_{j-1}^n)]\\&=\frac{1}{2}\left[\Delta t^2\nu^2(1-2\theta)-\frac{\nu\Delta t\Delta x^2}{6}\right]\left(\frac{\partial^4 u}{\partial x^4}\right)_j^n+O(\Delta t\Delta x^2,\Delta t^3,\Delta t^2\Delta x^2),\end{aligned}$$

故一般情况下格式具有 $\Delta t\Delta x^2$ 量级的精度. 当

$$\theta = \frac{1}{2}-\frac{1}{12\sigma} = \frac{1}{2}\left(1-\frac{1}{6\sigma}\right) \tag{2-32}$$

时，有较高的精度. 当 $\theta=\frac{1}{2}$ 时，称上述格式为 Crank-Nicolson 格式；当 $\theta=\frac{1}{2}\left(1-\frac{1}{6\sigma}\right)$ 时，上述格式称为 Mitchell-Fairweather 格式.

4. DuFort-Frankel 格式

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} = \nu\frac{u_{j+1}^n-u_j^{n+1}-u_j^{n-1}+u_{j-1}^n}{\Delta x^2}. \tag{2-33}$$

用冯·诺伊曼方法分析，这里设 $\varepsilon_j^n\sim A^n e^{ikx_j}$，可得

$$\begin{cases}(1+\sigma)A^{n+1}-2\sigma\cos(k\Delta x)A^n = (1-\sigma)A^{n-1},\\ A^n = A^n.\end{cases}$$

写成矩阵形式为

$$\begin{bmatrix} 0 & 1 \\ \dfrac{1+\sigma}{1-\sigma} & \dfrac{2\sigma\cos(k\Delta x)}{1-\sigma} \end{bmatrix} \begin{bmatrix} A^{n+1} \\ A^n \end{bmatrix} = \begin{bmatrix} A^n \\ A^{n-1} \end{bmatrix},$$

放大矩阵

$$G = \begin{bmatrix} 0 & 1 \\ \dfrac{1+\sigma}{1-\sigma} & \dfrac{2\sigma\cos(k\Delta x)}{1-\sigma} \end{bmatrix}^{-1} = \begin{bmatrix} -\dfrac{2\sigma\cos(k\Delta x)}{1+\sigma} & \dfrac{1-\sigma}{1+\sigma} \\ 1 & 0 \end{bmatrix}.$$

为使 $\| G \| \leqslant 1$,要求 G 的两个特征值的绝对值均不大于 1. G 的特征值由 $\det | G - \lambda I | = 0$ 确定,

$$\lambda = \frac{\sigma\cos(k\Delta x) \pm \sqrt{1 - \sigma^2\sin^2(k\Delta x)}}{1+\sigma}.$$

无论 $\sigma^2\sin^2(k\Delta x) > 1$ 或是 $\sigma^2\sin(k\Delta x) < 0$,都不难证明 $| \lambda | \leqslant 1$,故本格式无条件稳定.但用泰勒展开可得截断误差为 $O(\Delta x^2, \Delta t^2, \Delta t/\Delta x)$,故必须 $\Delta t/\Delta x \to 0$ 时格式才是相容的,因此限制了本格式的使用.

2.3.2 对流方程(双曲型方程,$\nu = 0$)

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0. \tag{2-34}$$

1. 迎风(Godunor)格式

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{a + |a|}{2}\frac{u_j^n - u_{j-1}^n}{\Delta x} + \frac{a - |a|}{2}\frac{u_{j+1}^n - u_j^n}{\Delta x} = 0. \tag{2-35}$$

用冯·诺伊曼方法得放大因子

$$G = 1 - s + s e^{-ik\Delta x}, \quad s = \frac{|a|\Delta t}{\Delta x}. \tag{2-36}$$

可见稳定条件为 $s = \dfrac{|a|\Delta t}{\Delta x} \leqslant 1$.该数称为库兰特(Courant)数,其稳定条件叫做 CFL 条件(Courant Friedvich Lewy).迎风格式在时间和空间方向均为一阶的.

2. 弗里德里希-拉克斯格式

由于方程(2-34)的特征线为 $x - at = \text{const}$,沿特征线上的 u 是不变的,所以 u_j^{n+1} 应当与 u_A^n 相等,其中 A 点是过(x_j, t_{n+1})点的直线 $x - at = x_j - at_{n+1}$ 与直线 $t = t_n$ 的交点.当 $a > 0$ 时,A 点用 u_{j-1}^n、u_j^n 插值得到,当 $a < 0$ 时 A 点用 u_j^n、u_{j+1}^n 插值得到,这就是上述迎风格式的由来.若 A 点值直接由 u_{j-1}^n、u_{j+1}^n 之间线性插值得到,就构成了弗里德里希-拉克斯格式:

$$u_j^{n+1} = u_{j-1}^n + \frac{\Delta x - a\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n),$$

即

$$u_j^{n+1} = \frac{1}{2}(1+s)u_{j-1}^n + \frac{1}{2}(1-s)u_{j+1}^n. \tag{2-37}$$

放大因子

$$G(k, \Delta x) = \cos(k\Delta x) - i\sin(k\Delta x),$$

稳定条件为

$$|s| = \left|\frac{a\Delta t}{\Delta x}\right| \leqslant 1.$$

3. 欧拉隐式格式

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+a\frac{u_{j+1}^{n+1}-u_{j-1}^{n+1}}{2\Delta x}=0. \tag{2-38}$$

放大因子

$$G=\frac{1}{\sqrt{1+s^2\sin^2(k\Delta x)}}\leqslant 1.$$

故欧拉格式为无条件稳定.顺便指出显式欧拉格式为无条件不稳定的.

4. 蛙跳式(leap-frog)格式

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}+a\frac{u_{j+1}^{n}-u_{j-1}^{n}}{2\Delta x}=0. \tag{2-39}$$

放大因子矩阵

$$G=\begin{bmatrix}-\mathrm{i}2s\sin(k\Delta x) & 1\\ 1 & 0\end{bmatrix}.$$

该矩阵特征值的模小于1的条件为

$$s=\frac{|a|\Delta t}{\Delta x}\leqslant 1, \tag{2-40}$$

这就是稳定条件.显然蛙跳式格式在时间和空间方向都是二阶精度的.

5. 拉克斯-温德罗夫格式

如上所述,迎风和 Friedrich 格式都是采用 $u_j^{n+1}=u_A$ 的条件,只是因 u_A 的算法不同,从而构成了不同的格式.在本格式中 u_A 由 u_{j-1}^n、u_j^n、u_{j+1}^n 通过二次插值得到.由于这时 u_A 即 $u_j^n-\frac{s}{2}(u_{j+1}^n-u_{j-1}^n)+\frac{s^2}{2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)$,故本格式为

$$u_j^{n+1}=u_j^n-\frac{1}{2}s(u_{j+1}^n-u_{j-1}^n)+\frac{1}{\pi}s^2(u_{j+1}^n-2u_j+u_{j-1}^n). \tag{2-41}$$

它的放大因子为

$$G(k,\Delta x)=1-\mathrm{i}\sin(k\Delta x)-2s^2\sin^2(\frac{k\Delta x}{2}).$$

显然使 $|G|\leqslant 1$ 的条件为

$$s=\frac{|a|\Delta t}{\Delta x}\leqslant 1. \tag{2-42}$$

这就是拉克斯-温德罗夫格式的稳定条件.

6. Mac Cormack 格式

上一格式可改写为

$$\begin{cases}\overline{u^{n+1}}=u_j^n-s(u_{j+1}^n-u_j^n),\\ \overline{\overline{u^{n+1}}}=\overline{u_j^{n+1}}-s(\overline{u_j^{n+1}}-\overline{u_{j-1}^{n+1}}),\\ u_j^{n+1}=\frac{1}{2}(u_j^n+\overline{\overline{u_j^{n+1}}}),\end{cases}$$

考虑到 $s=\frac{a\Delta t}{\Delta x}$,又记 $au=f$,则上式可改写为

$$
\begin{cases}
\overline{u_j^{n+1}} = u_j^n - \dfrac{\Delta t}{\Delta x}(f_{j+1}^n - f_j^n), \\
\overline{\overline{u_j^{n+1}}} = \overline{u_j^{n+1}} - \dfrac{\Delta t}{\Delta x}(\overline{f_j^{n+1}} - \overline{f_{j-1}^{n+1}}), \\
u_j^{n+1} = \dfrac{1}{2}(u_j^n + \overline{\overline{u_j^{n+1}}}).
\end{cases}
\tag{2-43}
$$

这就是 Mac Cormack 格式,其中第一步为预测步,第二步为校正步. 当 f 为 u 的线性函数时,也即 n 为常数时,则该格式与 L-W 格式是一致的,稳定条件也相同. 当 f 不是 n 的线性函数时,它对应的微分方程为

$$
\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{2-44}
$$

或

$$
\frac{\partial u}{\partial t} + a(u)\frac{\partial u}{\partial x} = 0, \quad a(u) = \frac{\partial f}{\partial u}. \tag{2-45}
$$

2.3.3　对流扩散方程(方程(2-25))

1. 双层格式

该格式建立了 n 和 $n+1$ 两个时间层的节点上各量间的一般关系:

$$
a_1 u_{j+1}^{n+1} + a_0 u_j^{n+1} + a_{-1} u_{j-1}^{n+1} = \alpha_1 u_{j+1}^n + \alpha_0 u_j^n + \alpha_{-1} u_{j-1}^n. \tag{2-46}
$$

利用泰勒展开可建立该格式与微分方程相容的条件为

$$
\begin{cases}
(\alpha_1 + \alpha_0 + \alpha_{-1}) - (a_1 + a_0 + a_{-1}) = 0, \\
a(\alpha_1 + \alpha_0 + \alpha_{-1}) = [(\alpha_1 - \alpha_{-1}) - (a_1 - a_{-1})]\dfrac{\Delta x}{\Delta t}, \\
\nu(\alpha_1 + \alpha_0 + \alpha_{-1}) = -\dfrac{1}{2}[(\alpha_1 + \alpha_{-1}) - (a_1 + a_{-1})]\dfrac{\Delta x}{\Delta t},
\end{cases}
\tag{2-47}
$$

或

$$
\begin{cases}
a_1 = \dfrac{1}{2}(2\sigma - s + 2)\alpha_1 + \dfrac{1}{2}(2\sigma - s)\alpha_0 + \dfrac{1}{2}(2\sigma - s)\alpha_{-1}, \\
a_0 = -2\sigma\alpha_1 + (1 - 2\sigma)\alpha_0 - 2\sigma\alpha_{-1}, \\
a_{-1} = \dfrac{1}{2}(2\sigma + s)\alpha_1 + \dfrac{1}{2}(2\sigma + s)\alpha_0 + \dfrac{1}{2}(2\sigma + s + 2)\alpha_{-1}.
\end{cases}
\tag{2-48}
$$

$$
s = \frac{a\Delta t}{\Delta x}, \quad \sigma = \frac{\nu\Delta t}{\Delta x^2}. \tag{2-49}
$$

放大因子为

$$
G(k,\Delta x) = \frac{a_0 + (a_1 + a_{-1})\cos(k\Delta x) + \mathrm{i}(a_1 - a_{-1})\sin(k\Delta x)}{\alpha_0 + (\alpha_1 + \alpha_{-1})\cos(k\Delta x) + \mathrm{i}(\alpha_1 - \alpha_{-1})\sin(k\Delta x)}. \tag{2-50}
$$

不失一般性选 $a_{-1} + a_0 + a_1 = \alpha_{-1} + \alpha_0 + \alpha_1 = 1$,且引入记号

$$
\begin{cases}
X = \dfrac{1}{2}(1 - \cos(k\Delta x)), \quad B = 16a_1a_{-1}, B' = 16\alpha_1\alpha_{-1} \\
C = 4[(a_1 - a_{-1})^2 - a_1 + a_{-1}], \quad C' = 4[(\alpha_1 - \alpha_{-1})^2 - \alpha_1 + \alpha_{-1}]
\end{cases}
$$

不难验证

$$
G\overline{G} = 1 - X\frac{(B' - B)X + (C' - C)}{B'X^2 + C'X + 1},
$$

其中 $\overline{G}$ 为 G 的共轭. 注意到 $0 \leqslant X \leqslant 1$, 使 $G\overline{G} = |G|^2 \leqslant 1$ 的条件为

$$\begin{cases}(B' - B)X + (C' - C) \geqslant 0, \\ B' - B + C' - C \geqslant 0, \quad C' - C \geqslant 0,\end{cases}$$

即

$$\begin{cases}(\alpha_1 - \alpha_{-1})^2 - (a_1 - a_{-1})^2 - (\alpha_1 + \alpha_{-1} - a_1 - a_{-1}) \geqslant 0, \\ 4(\alpha_1\alpha_{-1} - a_1 a_{-1}) + (a_1 - \alpha_{-1})^2 - (\alpha_1 - \alpha_{-1})^2 - (\alpha_1 + \alpha_{-1} - a_1 - a_{-1}) \geqslant 0.\end{cases} \tag{2-51}$$

这就是本格式稳定的条件.

2. FTCS 格式

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = \nu\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}, \tag{2-52}$$

整理得 $$u_j^{n+1} = \frac{1}{2}(2\sigma - s)u_{j+1}^n + (1 - 2\sigma)u_j^n + \frac{1}{2}(2\sigma + s)u_{j-1}^n. \tag{2-53}$$

这相当于上一格式中取

$$\begin{cases}\alpha_1 = \alpha_{-1} = 0, \quad \alpha_0 = 1, \\ a_1 = \frac{1}{2}(2\sigma - s), a_0 = 1 - 2\sigma, a_{-1} = \frac{1}{2}(2\sigma + s).\end{cases}$$

代入(2-51)式得稳定条件为

$$2\sigma - s^2 \geqslant 0, \quad -[(2\sigma)^2 - s^2] + 2\sigma - s^2 \geqslant 0,$$

即 $$\sigma \leqslant \frac{2}{R^2}, \quad \sigma \leqslant \frac{1}{2},$$

其中 $R = \frac{a\Delta x}{\nu}$ 叫做**格子雷诺数**. 上述条件写为

$$\sigma = \frac{\nu\Delta t}{\Delta x^2} \leqslant \min\left\{\frac{1}{2}, \frac{2}{R^2}\right\}. \tag{2-54}$$

3. BTCS 格式

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} = \nu\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} \tag{2-55}$$

或 $$\frac{1}{2}(s - 2\sigma)u_{j+1}^{n+1} + (1 + 2\sigma)u_j^{n+1} + \frac{1}{2}(-2\sigma - s)u_{j-1}^{n+1} = u_j^n. \tag{2-56}$$

类似分析得稳定条件为

$$\sigma^2 + 2\sigma \geqslant 0, \quad 4\sigma^2 + \sigma \geqslant 0.$$

由于 $\sigma > 0$, 故无条件稳定.

4. Crank-Nicolson 格式

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\left[\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x}\theta + \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}(1 - \theta)\right]$$
$$= \nu\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2}\theta + \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}(1 - \theta)\right], \tag{2-57}$$

整理得

$$\begin{cases} a_1 = -\frac{1}{2}(2\sigma - s)(1-\theta), \quad a_0 = (1+2\sigma)(1-\theta), \\ a_{-1} = \frac{1}{2}(2\sigma + s)(1-\theta), \\ a_1 = -\frac{\theta}{2}(2\sigma - s), a_0 = 1 + 2\sigma\theta, a_{-1} = -\frac{\theta}{2}(2\sigma + s). \end{cases} \tag{2-58}$$

稳定条件为

$$\begin{cases} 2(2\theta - 1)\sigma + 1 \geqslant 0, \\ (2\theta - 1)s^2 + 4\sigma \geqslant 0. \end{cases}$$

故 $\theta \geqslant \frac{1}{2}$ 时为无条件稳定；

$0 < \theta < \frac{1}{2}$ 时，稳定条件为 $\sigma \leqslant \frac{1}{2(1-2\theta)}, s^2 \leqslant \frac{2}{(1-2\theta)^2}$.

$\theta = \frac{1}{2}$ 时即为 Crank-Nicolson 格式.

5. 迎风格式

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{a}{2\Delta x}[(1-\varepsilon)(u_{j+1}^n - u_j^n) + (1+\varepsilon)(u_j^n - u_{j-1}^n)]$$
$$- \frac{\nu}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) = 0, \tag{2-59}$$

其中

$$\varepsilon = \mathrm{sgn}(a) = \begin{cases} 1, & a > 0, \\ 0, & a = 0, \\ -1, & a < 0. \end{cases}$$

同样得稳定条件

$$\sigma \leqslant \frac{1}{2 + |R|}, \quad s \leqslant 1, \quad R = \frac{a\Delta x}{\nu}. \tag{2-60}$$

值得注意的是，利用泰勒公式可得上述格式的近似方程为

$$\left(\frac{\partial u}{\partial t}\right)_j^n + a\left(\frac{\partial u}{\partial x}\right)_j^n - (\nu + |a|\Delta x)\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n = O(\Delta t, \Delta x^2).$$

可见这里粘性增加了 $|a|\Delta x$ 倍，这一部分又称格式粘性系数，它的出现使精度下降，稳定性得以改善. 格式粘性项在流场计算中必须密切注意.

6. Mac Cormack 格式

记

$$f = au - \nu\frac{\partial u}{\partial x}, \quad \Delta^+ u_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x}, \quad \Delta^- u_j^n = \frac{u_j^n - u_{j-1}^n}{\Delta x},$$

本格式为

$$\begin{cases} \overline{u_j^{n+1}} = u_j^n - \Delta t \Delta^- f_j^n, \\ \overline{\overline{u_j^{n+1}}} = \overline{u_j^{n+1}} - \Delta t \Delta^+ \overline{f_j^{n+1}}, \\ u_j^{n+1} = \frac{1}{2}(u_j^n + \overline{\overline{u_j^{n+1}}}). \end{cases} \tag{2-61}$$

同样分析可得稳定条件为

$$\sigma \leqslant \frac{1}{2+|R|} \text{ 和 } s \leqslant 1. \tag{2-62}$$

当 a、ν 是常数时，方程是线性的，即可得到相应的拉克斯 - 温德罗夫格式，这里从略.

7. 蛙跳式格式

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2\Delta x}=\nu\frac{u_{j+1}^{n-1}-2u_j^{n-1}+u_{j-1}^{n-1}}{\Delta x^2}. \tag{2-63}$$

这是一个三层格式，它的放大因子矩阵为

$$G=\begin{bmatrix}-\mathrm{i}2s\sin(k\Delta x) & 1-8\sigma\sin^2(\frac{k\Delta x}{2})\\ 1 & 0\end{bmatrix},$$

它的特征方程为

$$\lambda^2+\mathrm{i}2s\sin(k\Delta x)\cdot\lambda-1+8\sigma\sin^2(\frac{k\Delta x}{2})=0.$$

使 $|\lambda|\leqslant 1$ 的条件为

$$0<s\leqslant 1-2\sigma,\quad \sigma\leqslant\frac{1}{2+|R|}. \tag{2-64}$$

DuFort-Frankel 蛙跳格式为

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2\Delta x}=\nu\frac{u_{j+1}^n-u_j^{n+1}-u_j^{n-1}+u_{j-1}^n}{\Delta x^2}, \tag{2-65}$$

其稳定条件为

$$\sigma\leqslant 1,\quad s\geqslant\sigma. \tag{2-66}$$

蛙跳式上风格式为

$$\begin{aligned}&\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}+\frac{a+|a|}{2\Delta x}(u_j^n-u_{j-1}^n)+\frac{a-|a|}{2\Delta x}(u_{j+1}^n-u_j^n)\\&=\frac{\nu}{\Delta x^2}(u_{j+1}^{n-1}-2u_j^{n-1}+u_{j-1}^{n-1}),\end{aligned} \tag{2-67}$$

稳定条件为

$$\sigma\leqslant\frac{1}{4},\quad s>2\sigma. \tag{2-68}$$

蛙跳式上风 DuFort-Frankel 格式为

$$\begin{aligned}&\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t}+\frac{a+|a|}{2\Delta x}(u_j^n-u_{j-1}^n)+\frac{a-|a|}{2\Delta x}(u_{j+1}^n-u_j^n)\\&=\frac{\nu}{\Delta x^2}(u_{j+1}^n-u_j^{n+1}-u_j^{n-1}+u_j^n),\end{aligned} \tag{2-69}$$

稳定条件

$$\sigma\leqslant 1,\quad \sigma\leqslant\sqrt{2}s. \tag{2-70}$$

2.4　多维问题的差分格式

二维的对流扩散问题的基本方程为

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}+b\frac{\partial u}{\partial y}=\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right). \tag{2-71}$$

建立差分格式有以下几个途径：

1. 在 x、y 方向分别用以上一维问题中采用的格式

以扩散方程为例，

$$\frac{\partial u}{\partial t}=\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

用 Crank-Nicolson 格式得

$$\frac{1}{\Delta t}(u_{j,k}^{n+1}-u_{j,k}^{n})=\nu[\theta(\Delta_{xx}u_{j,k}^{n+1}+\Delta_{yy}u_{j,k}^{n+1})+(1-\theta)(\Delta_{xx}u_{j,k}^{n}+\Delta_{yy}u_{j,k}^{n})],$$

其中

$$\Delta_{xx}u_{j,k}=\frac{u_{j+1,k}-2u_{j,k}+u_{j-1,k}}{\Delta x^2},\quad \Delta_{yy}u_{j,k}=\frac{u_{j,k+1}-2u_{j,k}+u_{j,k-1}}{\Delta y^2}.$$

设 $\varepsilon_{j,k}^{n}\sim A^n e^{i(k_x x+k_y y)}$，则放大因子为

$$G=\frac{1-4(1-\theta)\left[\sigma_x\sin^2\left(\frac{k_x\Delta x}{2}\right)+\sigma_y\sin^2\left(\frac{k_y\Delta y}{2}\right)\right]}{1+4\theta\left[\sigma_x\sin^2\left(\frac{k_x\Delta x}{2}\right)+\sigma_y\sin^2\left(\frac{k_y\Delta y}{2}\right)\right]}.$$

记 $B=\sigma_x\sin^2\left(\frac{k_x\Delta x}{2}\right)+\sigma_y\sin^2\left(\frac{k_y\Delta y}{2}\right)$，则

$$G=\frac{1-4(1-\theta)B}{1+4\theta B}$$

可见，$\theta\geqslant\frac{1}{2}$ 时，无条件稳定；

$\theta<\frac{1}{2}$ 时，稳定条件为 $\nu\Delta t\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)\leqslant\frac{1}{2(1-2\theta)}$.

又以双曲方程为例，

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}+b\frac{\partial u}{\partial y}=0.$$

引入旋转坐标 $x'=\frac{1}{\sqrt{a^2+b^2}}(ax+by)$，$y'=\frac{1}{\sqrt{a^2+b^2}}(-bx+ay)$，可得

$$\frac{\partial u}{\partial t}+\sqrt{a^2+b^2}\frac{\partial u}{\partial x'}=0.$$

于是前面介绍的各种一维下双曲方程的差分格式都可以使用. 相应地

$$s'=\frac{\sqrt{a^2+b^2}\Delta t}{\Delta x'}=\frac{(a^2+b^2)\Delta t}{a\Delta x+b\Delta y}.$$

2. 交替方向法(ADI)

隐式格式的直接引用，在多维问题中将生成过于庞大的代数方程组，以至于无法求解. 交替方向法就是在某一方向上用隐式格式而其它方向上用显式，该方向轮换地取为各个方向.

以扩散方程为例,具体形式为

$$\begin{cases} \dfrac{u_{j,k}^{2n+1} - u_{j,k}^{2n}}{\Delta t} = \nu(\Delta_{xx}u_{j,k}^{2n+1} + \Delta_{yy}u_{j,k}^{2n}), \\ \dfrac{u_{j,k}^{2n+2} - u_{j,k}^{2n+1}}{\Delta t} = \nu(\Delta_{xx}u_{j,k}^{2n+1} + \Delta_{yy}u_{j,k}^{2n+2}), \end{cases}$$

放大因子为

$$G = G_1G_2 = \frac{1 - 4\sigma_y\sin^2\left(\dfrac{k_y\Delta y}{2}\right)}{1 + 4\sigma_x\sin^2\left(\dfrac{k_x\Delta x}{2}\right)} \cdot \frac{1 - 4\sigma_x\sin^2\left(\dfrac{k_x\Delta x}{2}\right)}{1 + 4\sigma_y\sin^2\left(\dfrac{k_y\Delta y}{2}\right)}.$$

故无条件稳定.但该结论不能直接推广到三维,在三维问题中可采用

$$\begin{cases} \dfrac{u_{j,k,l}^{*} - u_{j,k,l}^{n}}{\Delta t} = \nu(\Delta_{xx}u_{j,k,l}^{*} + \Delta_{yy}u_{j,k,l}^{n} + \Delta_{zz}u_{j,k,l}^{n}), \\ \dfrac{u_{j,k,l}^{**} - u_{j,k,l}^{*}}{\Delta t} = \nu(\Delta_{yy}u_{j,k,l}^{**} - \Delta_{yy}u_{j,k,l}^{*}), \\ \dfrac{u_{j,k,l}^{n+1} - u_{j,k,l}^{**}}{\Delta t} = \nu(\Delta_{zz}u_{j,k,l}^{n+1} - \Delta_{zz}u_{j,k,l}^{**}), \end{cases}$$

该格式也是无条件稳定的.

3. 时间分裂格式

利用泰勒展开

$$\begin{aligned} u_{j,k}^{n+1} &= u_{j,k}^{n} + \left(\frac{\partial u}{\partial t}\right)_{j,k}^{n}\Delta t + \frac{1}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_{j,k}^{n}\Delta t^2 + \cdots \\ &= u_{j,k}^{n} + (L_x + L_y)u_{j,k}^{n}\Delta t + (L_x + L_y)^2u_{j,k}^{n}\frac{\Delta t^2}{2} + \cdots \\ &= (1 + L_x)(1 + L_y)u_{j,k}^{n}\Delta t + O(\Delta t^2), \end{aligned}$$

其中 L_x、L_y 分别为 x、y 方向上的差分算子.在略去 Δt^2 以上小量后可得

$$\begin{cases} u_{j,k}^{*} = (1 + L_y)u_{j,k}^{n}, \\ u_{j,k}^{n+1} = (1 + L_x)u_{j,k}^{*}, \end{cases}$$

这就相当于将一个二维问题分裂成两个一维问题,于是求解大为简化.这个方法也可推广到三维问题中去.该方法最早是由 ЯМенко 提出的,该方法的一个缺点是边界条件处理有时比较困难.

2.5 修正方程及其应用

前面已经提到修正方程是差分方程所对应的微分方程,它是通过泰勒展开得到的,是差分方程求解时实际求解的微分方程.它逼近原微分方程,但有误差,可以说是原方程的修正,所以叫修正方程.下面再以双曲型方程为例说明之.

双曲型方程为

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = 0, & t > 0, \\ u(x,0) = e^{i\alpha x}, & -\infty < x < +\infty. \end{cases} \tag{2-72}$$

迎风格式为

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0, \quad a > 0. \tag{2-73}$$

若选 $\Delta x = a\Delta t$,则上式为

$$u_j^{n+1} = u_{j-1}^n,$$

所得的解和精确解一致.当 $\Delta x \neq a\Delta t$ 时,用半离散格式

$$\frac{\partial u}{\partial t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0,$$

它的解为

$$u(x,t) = e^{i\alpha\left(x - \frac{a\sin\alpha\Delta x}{\alpha\Delta x}\right) - a\left(1 - \frac{\cos(\alpha\Delta x)}{\Delta x}\right)t}.$$

与精确解相比,幅度以 $\exp\left(-a\dfrac{1-\cos(\alpha\Delta x)}{\Delta x}t\right)$ 速率下降,而波速下降为 $\dfrac{a\sin\alpha\Delta x}{\alpha\Delta x}$.只有当 Δx、$\Delta t \to 0$ 时,幅度才不下降,波速保持为 a.可见离散化后产生了附加的衰减及波速的变化.衰减就是耗散.波速变化与波数 α 有关,所以叫做色散.这种变化是由差分引起的.将(2-73)式改写为

$$u_j^{n+1} = (1-s)u_j^n + su_{j-1}^n, \quad s = \frac{a\Delta t}{\Delta x}. \tag{2-74}$$

利用泰勒展开得

$$u + \Delta t\frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + \frac{\Delta t^3}{3!}\frac{\partial^3 u}{\partial t^3} + \cdots$$
$$= (1-s)u + s\left[u - \Delta x\frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\Delta x^3}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots\right],$$

这里省略了标号 j、n.上式化简得

$$\left\{\frac{\partial}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2}{\partial t^2} + \frac{\Delta t^2}{3!}\frac{\partial^3}{\partial t^2} + \cdots\right\}u$$
$$= \left\{-a\frac{\partial}{\partial x} + \frac{a\Delta x}{2}\frac{\partial^2}{\partial x^2} - \frac{a\Delta x^2}{6}\frac{\partial^3}{\partial x^3} + \cdots\right\}u,$$

简写为

$$L_t u = L_x u.$$

由于 L_t、L_x 是线性算子,故有

$$L_t^l u = L_x^l u, \quad l = 1, 2, \cdots.$$

像前面一样,取 $l = 2, 3, \cdots$,可以逐次得到 $\dfrac{\partial^2 u}{\partial t^2}, \dfrac{\partial^3 u}{\partial t^3}, \dfrac{\partial^4 u}{\partial t^4}\cdots$ 用空间导数表示的表达式,整理后得

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \frac{a\Delta x}{2}(1-s)\frac{\partial^2 u}{\partial x^2} - a\Delta x^2(2s^2 - 3s + 1)\frac{\partial^3 u}{\partial x^3} + \cdots \tag{2-75}$$

或记作

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \sum_{k=1}^{\infty}\nu_{2k}\frac{\partial^{2k}u}{\partial x^{2k}} + \sum_{k=1}^{\infty}\nu_{2k+1}\frac{\partial^{2k+1}u}{\partial x^{2k+1}}, \tag{2-76}$$

这里 ν_{2k} 使方程增加耗散，而 ν_{2k+1} 使方程增加色散，其中

$$\nu_2 = \frac{a\Delta x}{2}(1-s), \quad \nu_3 = -a\Delta x^2(2s^2-3s+1).$$

再设 $u \sim e^{\beta t}e^{i\alpha x}$，代入(2-75)式得

$$\begin{cases}\beta = \sum_{k=1}^{\infty}(-1)^k\nu_{2k}\alpha^{2k},\\ a = i\sum_{k=1}^{\infty}(-1)^k\nu_{2k+1}\alpha^{2k+1}.\end{cases}$$

可得差分格式的稳定条件为

$$(-1)^k\nu_{2k} < 0, \quad k = 1,2,\cdots. \tag{2-77}$$

若采用 L-W 格式，则有

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = \frac{a^2\Delta t}{2}\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}, \tag{2-78}$$

其修正方程为

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = -\frac{a}{6}\Delta x^2(1-s^2)\frac{\partial^3 u}{\partial x^3} - \frac{a}{8}\Delta x^3 s(1-s^2)\frac{\partial^4 u}{\partial x^4} - \cdots. \tag{2-79}$$

另一种具有二阶精度的格式为

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = \frac{a}{2}(\Delta x - a\Delta t)\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}, \tag{2-80}$$

它的修正方程为

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \frac{a}{6}\Delta x^2(1-s)(2-s)\frac{\partial^3 u}{\partial x^3} - \frac{a}{8}\Delta x^4 s(1-s^2)(2-s)\frac{\partial^4 u}{\partial x^4}\Delta x^4 + \cdots. \tag{2-81}$$

记 L-W 格式为

$$u_j^{n+1} = \left(1 - \frac{s}{2}\Delta_0 + \frac{s^2}{2}\Delta^+\Delta^-\right)u_j^n = L(s)u_j^n, \tag{2-82}$$

(2-80) 式可写为

$$u_j^{n+1} = L(1-s)L(1)u_j^n.$$

将二者组合得

$$u_j^{n+1} = \frac{1}{2}[L(s) + L(1-s)L(1)]u_j^n. \tag{2-83}$$

它的修正方程是两个格式的修正方程(2-75) 和(2-79) 之和. 当 $s < 1$ 时，两个格式的色散项互相抵消，耗散略为提高，于是所得的格式更稳定而色散减少了. (2-82) 式中 $\Delta^+u_j^n = u_{j+1}^n - u_j^n$；$\Delta^-u_j^n = u_j^n - u_{j-1}^n$.

可见利用修正方程可以进行稳定性分析，得到稳定的充分条件. 利用修正方程还可以设计精度更高、色散小、稳定性好的格式. 事实上由(2-75) 式知，它对应的格

式是一阶精度的，为了提高精度，要将(2-75)式右端第一项$\frac{a\Delta x}{2}(1-s)\frac{\partial^2 u}{\partial x^2}$消去. 格式(2-80)正是这样得到的，所得格式的修正方程中果然没有了一阶项(见(2-81)式). 这种做法正是反扩散格式的基本思想.

可见修正方程可以用来分析稳定性，还可用来构造新的格式.

3 可压缩流体流动的差分格式

控制可压缩流体流动的基本方程是纳维-斯托克斯(Navier-Stokes方程，N-S方程). 与粘性相关的二阶导数项都是线性项，并不带来计算上新的困难. 在不计粘性的条件下，基本方程就简化为欧拉方程. 由于它是非线性的，具有双曲型特性，并且容易产生间断，所以正确地捕捉激波是可压缩流体流动计算中的核心问题.

3.1 间断、弱解、熵函数、熵通量和熵条件

双曲型守恒方程的一般形式为

$$\frac{\partial \boldsymbol{u}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}=0, \tag{3-1}$$

且有

$$\boldsymbol{A}=\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}. \tag{3-2}$$

$\boldsymbol{A}$的特征值全都是实数. 在气体动力学中由于$\boldsymbol{F}$各项都是$\boldsymbol{u}$各项的齐次二次多项式，所以，还存在

$$\boldsymbol{F}=\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\boldsymbol{u}=\boldsymbol{A}\boldsymbol{u}. \tag{3-3}$$

不难得知存在特征线和特征关系式(见第1章). 当同族特征线相交时就意味着间断的产生. 在间断处方程的解不连续，所以方程本身不能满足，但主导流动的守恒律仍然满足，故在间断处作封闭曲线，在该曲线所围之面积上对(3-1)式积分，利用高斯积分公式，可得

$$[u]D=[\boldsymbol{F}],$$

其中[·]表示间断两侧值的差，D为间断线的斜率$\frac{\mathrm{d}x}{\mathrm{d}t}$. 由于存在间断，所以微分方程在间断处不再得到满足，为此需要引入弱解的概念.

所谓弱解是指满足与微分方程对应的积分方程的解. 因此可以说：在连续可微处满足微分方程，在不连续处满足间断条件的解称为弱解.

弱解也可以这样来定义：设φ为试验函数，它在$t>0$的半平面内的有界区域Ω内无穷可微，在Ω外为零；$\boldsymbol{u}$是分片连续可微的函数，对任意试验函数φ均有

$$\iint\limits_{\substack{t>0\\ \Omega}}\left[\boldsymbol{u}\frac{\partial\varphi}{\partial t}+f\frac{\partial\varphi}{\partial x}\right]\mathrm{d}x\mathrm{d}t=0$$

成立;则称 u 是微分方程的弱解.可以证明这两个弱解的定义是等价的.

大家知道,满足间断条件的解不一定都是方程的解,只有同时还满足熵条件的解才是真正有意义的解.在气体力学中,流体微团的熵函数就是熵本身,对于一般的非线性双曲型方程,需人为地构造熵函数和相应的熵通量.

熵函数记为 $U(u)$,相应的熵通量记为 $F(u)$,它们需要满足以下条件:

1° 相容性条件:

$$F_u = U_u f_u, \tag{3-4}$$

其中

$$\begin{cases} F_u = \dfrac{\partial F}{\partial u} = \left(\dfrac{\partial F}{\partial u_1}, \dfrac{\partial F}{\partial u_2}, \cdots, \dfrac{\partial F}{\partial u_n}\right) \\ U_u = \dfrac{\partial U}{\partial u} = \left(\dfrac{\partial U}{\partial u_1}, \dfrac{\partial U}{\partial u_2}, \cdots, \dfrac{\partial U}{\partial u_n}\right) \end{cases} \tag{3-5}$$

2° U 是 u 的严格的凸函数,即

$$U(\alpha u_1 + (1-\alpha)u_2) \leqslant \alpha U(u_1) + (1-\alpha)U(u_2)$$

$$0 \leqslant \alpha \leqslant 1$$

在这种情况下,黑塞(Hesse)矩阵

$$U_{uu} = \frac{\partial^2 U}{\partial u \partial u} = \left[\frac{\partial^2 U}{\partial u_i \partial u_j}\right] \tag{3-6}$$

是正定的.

不难看出,在连续可微区内

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0,$$

即熵守恒.对于间断区,则有熵不等式

$$[U(u)]\frac{\mathrm{d}x}{\mathrm{d}t} - [F(u)] \geqslant 0$$

或

$$\oint U(u)\mathrm{d}x - F(u)\mathrm{d}t \geqslant 0. \tag{3-7}$$

可以证明这是允许弱解的充要条件,另外一个表达形式可以是

$$\int_{\substack{t>0 \\ \Omega}} \left(U(u)\frac{\partial \varphi}{\partial t} + F(u)\frac{\partial \varphi}{\partial x}\right)\mathrm{d}x\mathrm{d}f \geqslant 0, \tag{3-8}$$

这个条件也叫解析熵条件,和前者是等价的.熵函数和熵通量合在一起叫做熵对.熵对构造如下:

在单一方程时,即变量 u 是一数量时,则可设

$$U = \frac{1}{2}u^2, \quad F = u f(u) - \int f(u)\mathrm{d}u. \tag{3-9}$$

在方程组时,设矩阵 $A = \dfrac{\partial f}{\partial u}$ 的特征为 $\lambda_1, \lambda_2, \cdots, \lambda_n$,组成对角线阵 $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$,且有

$$A = R\Lambda R^{-1} = L^{-1}\Lambda L = R\Lambda L, \tag{3-10}$$

其中 R 由右向量列组成,L 由左向量列组成.于是

$$\begin{cases} U(u) = u^{\mathrm{T}}L^{\mathrm{T}}LU, \\ F(u) = uL^{\mathrm{T}}\Lambda Lu. \end{cases} \tag{3-11}$$

不难证明它满足熵对条件.

设矩阵 A 的特征由小到大地排列为

$$\lambda_1 < \lambda_2 < \cdots < \lambda_n, \tag{3-12}$$

若间断的斜率$\frac{\mathrm{d}x}{\mathrm{d}t}$介于$\lambda_k(u_R)$和$\lambda_k(u_L)$之间,即

$$\lambda_k(u_R) < s < \lambda_k(u_L) \tag{3-13}$$

则称间断为 k 激波间断.这意味着左右两侧同为 k 族特征线相交(图 3-1).

一个简单而又清楚的熵条件是Олейник熵条件,它是对单个变量方程而言的:

$$\frac{f(u_R)-f(u)}{u_R-u} < \frac{f(u_R)-f(u_L)}{u_R-u_L} < \frac{f(u)-f(u_L)}{u-u_L}, \tag{3-14}$$

$$u \subset (u_R, u_2)$$

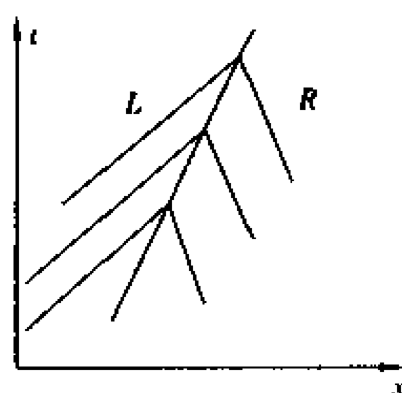


图 3-1

业已证明(3-1)式中 u 为一单变量 u 时,满足熵条件(3-14)的弱解是唯一的.

应当指出,条件(3-13)确定的满足熵条件的弱解更加严格.条件(3-14)对于标量方程才是成立的,并且它与条件(3-7)是等价的,对方程组尚不能证明一定等价.对于方程组,拉克斯提出了另一个与条件(3-13)等价的条件:

$$\lambda_{k-1}(u_L) < s < \lambda_k(u_L), \quad \lambda_k(u_R) < s < \lambda_{k+1}(u_R). \tag{3-15}$$

3.2 伯格方程及其求解

伯格(Burger)方程是一个模型方程,具有非线性对流扩散方程的一些主要特性,而本身比较简单,便于分析,所以许多新的差分格式均以它为首选加以应用,以验证格式的捕捉激波能力.伯格方程为

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}. \tag{3-16}$$

引入 Hopf-Cole 变换:

$$u = -2\nu\frac{\psi_x}{\psi} = -2\nu(\ln\psi)_x \tag{3-17}$$

可使(3-16)式改写为

$$\left(\frac{\psi_x}{\psi}\right)_t = \nu\left(\frac{\psi_{xx}}{\psi}\right)_x. \tag{3-18}$$

计及$(\psi_x/\psi)_t = (\psi_t/\psi)_x$,上式可改写为

$$\left(\frac{\psi_t}{\psi}\right)_x = \nu\left(\frac{\psi_{xx}}{\psi}\right)_x.$$

对 x 积分，计及在 $\pm\infty$ 处 $u=\text{const}$，故有

$$\psi_t=\nu\psi_{xx},\tag{3-19}$$

这是一个典型的热传导方程．积分得

$$\begin{cases}\psi(x,t)=\dfrac{1}{\sqrt{4\pi\nu t}}\displaystyle\int_{-\infty}^{\infty}\Phi(\xi)\exp\left\{-\frac{(x-\xi)^2}{4\nu t}\right\}\mathrm{d}\xi,\\ \Phi(x)=\psi(x,0)=\exp\left\{-\dfrac{1}{2\nu}\displaystyle\int_0^x\varphi(\eta)\mathrm{d}\eta\right\},\\ \varphi(x)=u(x,0),\end{cases}$$

进而得

$$u(x,t)=\frac{\displaystyle\int_{-\infty}^{\infty}\frac{x-\xi}{t}\exp\left\{-\frac{1}{2\nu}\int_0^{\xi}\varphi(\eta)\mathrm{d}\eta-\frac{(x-\xi)^2}{4\nu t}\right\}\mathrm{d}\xi}{\displaystyle\int_{-\infty}^{\infty}\exp\left\{-\frac{1}{2\nu}\int_0^{\xi}\varphi(\eta)\mathrm{d}\eta-\frac{(x-\xi)^2}{4\nu t}\right\}\mathrm{d}\xi}.\tag{3-20}$$

若

$$u(x,0)=\varphi(x)=\begin{cases}1, & x<0\\ 0, & x=0,\\ -1, & x>0.\end{cases}\tag{3-21}$$

则

$$u(x,t)=\frac{-\exp\left(\dfrac{x}{\nu}\right)\displaystyle\int_{-\frac{x+t}{2\sqrt{\nu t}}}^{\infty}\exp\{-\xi^2\}\mathrm{d}\xi+\int_{\frac{x-t}{2\sqrt{\nu t}}}^{\infty}\exp\{-\xi^2\}\mathrm{d}\xi}{\exp\left(\dfrac{x}{\nu}\right)\displaystyle\int_{-\frac{x+t}{2\sqrt{\nu t}}}^{\infty}\exp\{-\xi^2\}\mathrm{d}\xi+\int_{\frac{x-t}{2\sqrt{\nu t}}}^{\infty}\exp\{-\xi^2\}\mathrm{d}\xi}.\tag{3-22}$$

当 $t\to\infty$ 时有

$$u(x,t)=-\tanh\left(\frac{x}{2\nu}\right).\tag{3-23}$$

不难看出，当 $\nu\to0$ 时 $u(x,t)\to\varphi(x)$，即间断保持不变．

当 $\nu=0$ 时方程(3-16)成为无粘伯格方程：

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=0,\tag{3-24}$$

它的解在 $u(x,0)=\varphi(x)$ 初始条件下为

$$u(x,t)=\varphi(z),\quad z=x-u(x,t)t.\tag{3-25}$$

不难看出

$$\frac{\partial u(x,t)}{\partial x}=\varphi'\left(1-\frac{\partial u}{\partial x}t\right)$$

或

$$\frac{\partial u}{\partial x}=\frac{\varphi'(z)}{1+t\varphi'(z)},$$

当 $\varphi'(z)<0$ 及 $-t\varphi'(z)=1$ 时，$\dfrac{\partial u}{\partial x}\to\infty$．如图3-2所示

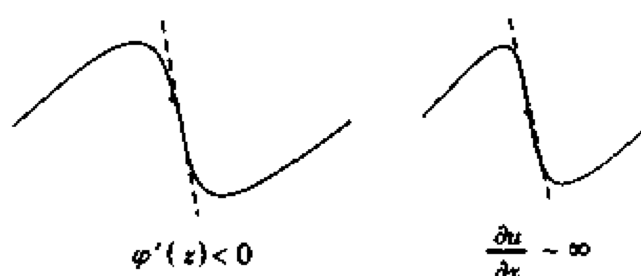


图 3-2

这就意味着连续点的出现.该方程的特征为

$$\lambda = \frac{\mathrm{d}x}{\mathrm{d}t} = u, \tag{3-26}$$

间断条件为

$$[u]D = \left[\frac{u^2}{2}\right], \tag{3-27}$$

熵条件为

$$\frac{f(u_L)-f(u)}{u_L-u} \geqslant \frac{f(u_R)-f(u_L)}{u_R-u_L} \geqslant \frac{f(u_R)-f(u_L)}{u_R-u_L}. \tag{3-28}$$

计及 $f(u)=\frac{u^2}{2}$,则由上式得

$$u_L > u > u_R, \tag{3-29}$$

即只有在 $u_L > u_R$ 时才可能出现间断,因为它满足熵条件.下面考察一些早期格式用于求解起始条件为(3-21)式时的无粘伯格方程的数值解

1. 迎风格式(Godunov 格式)

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + \frac{u_j^n+|u_j^n|}{2|u_j^n|}\frac{f_j^n-f_{j-1}^n}{\Delta x} + \frac{u_j^n-|u_j^n|}{2|u_j^n|}\frac{f_{j+1}^n-f_j^n}{\Delta x} = 0. \tag{3-30}$$

稳定条件为

$$|u|_{\max}\frac{\Delta t}{\Delta x} \leqslant 1, \tag{3-31}$$

它是启发性的结果,只作参考,因为方程是非线性的.下面也是一样.

2. 拉克斯格式

$$u_j^{n+1} = \frac{u_{j+1}^n+u_{j-1}^n}{2} - \frac{\Delta t}{\Delta x}\frac{f_{j+1}^n-f_{j-1}^n}{2}. \tag{3-32}$$

稳定条件为

$$|u|_{\max}\frac{\Delta t}{\Delta x} \leqslant 1. \tag{3-33}$$

3. 拉克斯 - 温德罗夫格式

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\frac{f_{j+1}^n-f_{j-1}^n}{2} + \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2\left[A_{j+\frac{1}{2}}^n(f_{j+1}^n-f_j^n) - A_{j-\frac{1}{2}}^n(f_j^n+f_{j-1}^n)\right]. \tag{3-34}$$

稳定条件也是

$$|u|_{\max}\frac{\Delta t}{\Delta x} \leqslant 1.$$

4. S_β^α 格式

$$\begin{cases}\overline{u_j^{n+1}} = (1-\beta)u_j^n + \beta u_{j+1}^n - \alpha\dfrac{\Delta t}{\Delta x}(f_{j+1}^n - f_j^n),\\ u_j^{n+1} = u_j^n - \dfrac{\Delta t}{2\alpha\Delta x}\left[(\alpha-\beta)f_{j+1}^n + (2\beta-1)f_j^n + (1-\alpha-\beta)f_{j-1}^n + \overline{f_j^{n+1}} - \overline{f_{j-1}^{n+1}}\right].\end{cases} \tag{3-35}$$

当 $\alpha = 1, \beta = 0$ 或 1 时，为 Mac Cormack 格式；

当 $\alpha = \dfrac{1}{2}, \beta = \dfrac{1}{2}$ 时，为 Richitmyer 格式；

当 $\alpha = 1, \beta = \dfrac{1}{2}$ 时，为 Rulin-Burstein 格式；

当 $0 < \alpha < 1, \beta = \dfrac{1}{2}$ 时，为 MeGuire-Morris格式；

其中 $\overline{u^{n+1}}$ 的预测值是图 3-3 中虚线交点的值，所以稳定条件也是

$$|u|_{\max}\frac{\Delta t}{\Delta x} \leqslant 1. \tag{3-36}$$

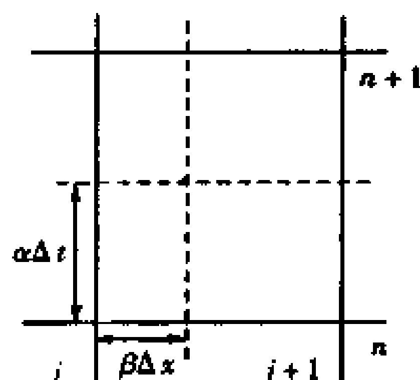


图 3-3

5. Beam-Warming 显式格式

$$\begin{cases}\overline{u_j^{n+1}} = \begin{cases}u_j^n - \dfrac{\Delta t}{\Delta x}(f_j^n - f_{j-1}^n), & u_j^n > 0,\\ u_j^n - \dfrac{\Delta t}{\Delta x}(f_{j+1}^n - f_j^n), & u_j^n < 0,\end{cases}\\ u_j^{n+1} = \begin{cases}u_j^n - \dfrac{\Delta t}{2}\left[\dfrac{2f_j^n - 3f_{j-1}^n + f_{j-2}^n}{\Delta x} + \dfrac{\overline{f_j^{n+1}} - \overline{f_{j-1}^{n+1}}}{\Delta x}\right], & u_j^n > 0,\\ u_j^n - \dfrac{\Delta t}{2}\left[\dfrac{-2f_j^n + 3f_{j+1}^n - f_{j+2}^n}{\Delta x} + \dfrac{\overline{f_{j+1}^{n+1}} - \overline{f_j^{n+1}}}{\Delta x}\right], & u_j^n < 0.\end{cases}\end{cases} \tag{3-37}$$

稳定条件为

$$|u|_{\max}\frac{\Delta t}{\Delta x} \leqslant 2.$$

6. Beam-Warming 隐式格式

它由 $\dfrac{u_j^{n+1} - u_j^n}{\Delta t} = -\dfrac{1}{2}\left[\left(\dfrac{\partial f}{\partial x}\right)_j^{n+1} + \left(\dfrac{\partial f}{\partial x}\right)_j^n\right]$ 得到

$$\frac{\Delta t}{2\Delta x}a_{j-1}^n u_{j-1}^{n+1} + u_j^{n+1} + \frac{\Delta t}{4\Delta x}a_{j+1}^n u_{j+1}^{n+1}$$
$$= -\frac{\Delta t}{2}\frac{f_{j+1}^n - f_{j-1}^n}{\Delta x} - \frac{\Delta t}{4\Delta x}a_{j-1}^n u_{j-1}^n + u_j^n + \frac{\Delta t}{4\Delta x}a_{j+1}^n u_{j+1}^n \tag{3-38}$$

它是无条件中性稳定的.

7. Rusanov I 格式

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}(f_{j+1}^n - f_{j-1}^n) + \frac{\omega}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{3-39}$$

其中 ω 为选用参数. 如用下面格式，则有三阶精度：

$$\begin{cases} u_{j+\frac{1}{2}}^{(1)} = \frac{1}{2}(u_{j+1}^n + u_j^n) - \frac{\Delta t}{3\Delta x}(f_{j+1}^n - f_j^n), \\ u_j^{(2)} = u_j^n - \frac{2}{3}\frac{\Delta t}{\Delta x}(f_{j+\frac{1}{2}}^{(1)} - f_{j-\frac{1}{2}}^{(1)}), \\ u_j^{n+1} = u_j^n - \frac{1}{24}\frac{\Delta t}{\Delta x}(-2f_{j+2}^{(2)} + 7f_{j+1}^{(2)} - 7f_{j-1}^{(2)} + 2f_{j-2}^{(2)}) - \\ \qquad \frac{3\Delta t}{8\Delta x}(f_{j+1}^{(2)} - f_{j-1}^{(2)}) - \frac{\omega}{24}(u_{j+2}^n - 4u_{j+1}^n + 6u_j^n - 4u_{j-1}^n + u_{j-2}^n). \end{cases}$$

它也称 RusanovⅡ 格式.其稳定性条件为

$$\begin{cases} s = |u|_{\max}\frac{\Delta t}{\Delta x} \leqslant 1, \\ 4s^2 - s^4 \leqslant \omega \leqslant 3. \end{cases} \tag{3-40}$$

计算表明,迎风格式、拉克斯格式将间断抹平了;RusanovⅠ 格式和 Beam-Warming显式格式结果尚好;其它格式在间断附近产生强列振荡,甚至无法计算下去,为分析产生振荡的原因,可以采用修正方程的方法加以分析.

修正方程的统一形式为

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \frac{\partial}{\partial x}(\nu_2\frac{\partial u}{\partial x}) + \frac{\partial}{\partial x}(\nu_3\frac{\partial^2 u}{\partial x^2}) + \cdots, \tag{3-41}$$

其中 $\nu_2, \nu_3, \cdots$ 都可与 u 有关.在 u 变化比较平稳处,令 $u = u_0 + u'$,其中 u_0 为常数,u' 为一小量,代入上式,并将 u' 再写为u,$\nu'_2, \nu'_3\cdots$,也仍写为 ν_2, ν_3,它们是在 $u = u_0$ 处的值.略去高阶小量,得

$$\frac{\partial u}{\partial t} + u_0\frac{\partial u}{\partial x} = \nu_2\frac{\partial^2 u}{\partial x^2} + \nu_3\frac{\partial^3 u}{\partial x^3} + \cdots. \tag{3-42}$$

如迎风格式的修正方程为

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = & \left\{\frac{\Delta x}{2}(1 - 2\sigma u) + \Delta x^3(\frac{3}{4}\sigma u_x - \frac{3}{2}\sigma^2 u u_x)\right\}u_x^2 + \\ & \left\{\frac{\Delta x}{2}u(1 - \sigma u) + \Delta x^2[(5\sigma u - \frac{1}{2})u_x - \frac{5}{2}\sigma^2 u^2 u_x]\right\}u_{xx} + \\ & \Delta x^2 u\left\{(\frac{\sigma u}{2} - \frac{1}{6}) - \frac{1}{3}\sigma^2 u^2\right\}u_{xxx} + \cdots. \end{aligned} \tag{3-43}$$

设上游 $u_0 = 1$,则得

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\Delta x}{2}(1 - \sigma)u_{xx} - \frac{\Delta x^2}{6}(1 - \sigma)(1 - 2\sigma)u_{xxx} + \cdots. \tag{3-44}$$

可见与 $a = 1$ 时的线性方程的上风格式推导得到的修正方程是一致的.所以以后各格式的修正方程可以直接用相应的线性方程所得的修正方程,使推导大为简化.

由上式看到,$\sigma < 1$ 时衰减比较大,而色散项的作用是使解内不同波数的波随时间推移而相互错开.本来没有波动,是由于不同波数波相互叠加的结果,但随时间推移后,它们互相错位了,因此显示出波动了.当衰减足够大时,这些波不同程度地衰减,特别是高波数波衰减得比较快,因此使波动得到抑制甚至消失.可以看出迎风格式就是因为衰减大,即耗散项大而使解没有振荡,甚至将应有的间断被抹平

了.

拉克斯格式的修正方程为

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=\frac{\Delta x}{2}\left(\frac{1}{\sigma}-\sigma\right)u_{xx}+\frac{\Delta x^2}{3}(1-\sigma^2)u_{xxx}+\cdots. \tag{3-45}$$

在 $\sigma<1$ 时耗散项比迎风格式还大.

L-W 和 Mac Cormack 格式的修正方程为

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=-\frac{\Delta x^2}{6}(1-\sigma^2)\frac{\partial^3 u}{\partial x^3}-\frac{1}{8}\Delta x^3\sigma(1-\sigma)\frac{\partial^4 u}{\partial x^4}+\cdots, \tag{3-46}$$

在 $\sigma<1$ 时耗散是三阶的,所以不足以使振荡消失;又由于 $\sigma<1$ 时 $\nu_3=-\frac{\Delta x^2}{6}(1-\sigma^2)<0$,它使波的速度减少,所以使波向上游移.类似地 $u_0=-1$ 时会使波向下游移动.因此在间断附近,这些不同波数的波会分别向上下游移开,于是在间断前后都出现急剧的振荡.

Beam-Warming 显式格式的修正方程为

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=-\frac{1}{6}\Delta x^2(\sigma-1)(2-\sigma)\frac{\partial^3 u}{\partial x^3}-\frac{\Delta x^3}{8}(\sigma-1)^2(2-\sigma)\frac{\partial^4 u}{\partial x^4}+\cdots, \tag{3-47}$$

显然 $\sigma<2$ 时有三阶衰减.而 $\nu_3=-\frac{\Delta x^2}{6}(\sigma-1)(2-\sigma)$ 在 σ 比较小时大于0,所以使波速增加,所以应在间断下游出现波动.这与计算的结果是吻合的. Beam-Warming 隐式格式是中性稳定的,说明衰减为零,这时振荡不可避免,需加入加粘性项加以抑制.

Rusanov Ⅰ 由于加强了粘性,使间断抹平.而 Rusanov Ⅱ 格式的修正方程为

$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=-\frac{\Delta x^3}{24}\left(\frac{\omega}{\sigma}-4\sigma+\sigma^3\right)\frac{\partial^4 u}{\partial x^4}+\frac{\Delta x^4}{120}(-5\omega+4+15\sigma^2-4\sigma^4)\frac{\partial^5 u}{\partial x^5}+\cdots, \tag{3-48}$$

稳定条件要求 $\sigma<1, 4\sigma^2-\sigma^4\leqslant\omega\leqslant 3$,这就有 $\nu_4<0$.但由于 ν_4 是 $O(\Delta x^4)$ 的,所以不足以抑制振荡.根据以上分析可以看出,高精度可以减少衰减,但振荡也可能增加.为此人们提出以下的一些方法减少振荡.

1. 保单格式

一般显式格式可以写为如下的形式:

$$u_j^{n+1}=\sum_{k=-k_1}^{k_2}C_k u_{j+k}^n. \tag{3-49}$$

若 $u_{j+1}^n\geqslant u_j^n$,必导致 $u_{j+1}^{n+1}\geqslant u_j^{n+1}$, $\forall j$ 或 $u_{j+1}^n\leqslant u_j^n$,必导致 $u_{j+1}^{n+1}\leqslant u_j^{n+1}$, $\forall j$,则(3-48) 式称为保单格式.容易证明:

当 $C_k>0(\forall k)$ 时,(3-47) 式为保单格式.这是保单格式的充分条件.保单格式总是一阶精度的,因为

$$u_j^{n+1}-\sum_{l=-k_1}^{k_2}C_l u_{j+l}^n=\left(1-\sum_{l=-k_1}^{k_2}C_l\right)u_j^n-\Delta x\left(\frac{a\Delta t}{\Delta x}+\sum_{l=-k_1}^{k_2}lC_l\right)\frac{\partial u}{\partial x}+$$

$$\frac{1}{2}\Delta x^2\left(\frac{a^2\Delta t^2}{\Delta x^2}-\sum_{l=-k_1}^{k_2}l^2C_l\right)\frac{\partial^2 u}{\partial x^2}+\cdots, \tag{3-50}$$

其中 $a=\frac{\partial f}{\partial u}$.考虑到差分格式与微分方程本身是相容的,故有

$$\begin{cases}1-\sum\limits_{l=-k_1}^{k_2}C_l=0\Rightarrow\sum\limits_{l}C_l=1,\\ \frac{a\Delta t}{\Delta x}+\sum\limits_{l=-k_1}^{k}lC_l=0\Rightarrow\sum\limits_{l=-k_1}^{k_2}lC_l=-\frac{a\Delta t}{\Delta x}.\end{cases} \tag{3-51}$$

由于 $C_l>0(\forall l)$,考虑到

$$\left(\sum_i a_ib_i\right)^2\leqslant\sum_i a_i^2\sum_i b_i^2, \tag{3-52}$$

故有

$$\left(\frac{a\Delta t}{\Delta x}\right)^2=\left(\sum_{l=-k_1}^{k_2}lC_l\right)^2=\left(\sum_{l=-k_1}^{k_2}l\sqrt{C_l}\sqrt{C_l}\right)^2$$

$$\leqslant\sum_{l=-k_1}^{k_2}l^2C_l\sum_{l=-k_1}^{k_2}C_l=\sum_{l=-k_1}^{k_2}l^2C_l. \tag{3-53}$$

为使差分格式有二阶精度,需要

$$\frac{a_2\Delta t^2}{\Delta x^2}-\sum_{l=-k_1}^{k_2}l^2C_l=0. \tag{3-54}$$

由于(3-52)式中的等号只有在 a_i/b_i 与 i 无关时才可成立,而这在(3-53)式中是不成立的,所以(3-52)式中等号不成立,所以(3-54)不成立.因此保单格式只能有一阶精度.但由于它对各种波动的衰减作用很大,因此不会产生振荡,而且还将间断抹得模糊起来.

2. 混合反扩散格式

由于精度的下降和振荡的产生主要集中在间断附近,所以将具有二阶精度的格式 L_2 和只有一阶精度的格式 L_1 组合起来,使组合参数在间断附近使 L_1 起主要作用,而在离间断较远处 L_2 起主要作用.具体形式为

$$u_j^{n+1}=Lu_j^n=[\theta L_1+(1-\theta)L_2]u_j^n, \tag{3-55}$$

其中
$$\theta=\left|\frac{|u_{j+1}-u_j|-|u_j-u_{j-1}|}{|u_{j+1}-u_j|+|u_j-u_{j-1}|}\right|^m\quad(m=1\sim2). \tag{3-56}$$

显然在有间断和急剧振荡时 θ 比较接近1,在平缓变化区 θ 接近0.这就达到以上目的了.而二阶精度格式是这样建立的:

由迎风格式得知

$$u_j^{n+1} = L_1 u_j^n = u_j^n - \frac{a\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) + \frac{|a|\Delta t}{2\Delta x}(u_{j+1}^n - 2u_j^n + u_{j-1}^n),$$

对应的二阶精度格式是通过抵消二阶误差项得到的，即得

$$u_j^{n+1} = L_2 u_j^n = u_j^n - \frac{a\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) + \frac{|a|\Delta t}{2\Delta x}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) - \frac{|a|\Delta t}{\Delta x}\left(1 - \frac{|a|\Delta t}{\Delta x}\right)(u_{j+1}^n - 2u_j^n + u_{j-1}^n). \tag{3-57}$$

这种建立二阶精度格式的方法叫做反扩散法.该方法可以扩展到隐式格式的建立，这时一阶隐式格式为

$$\begin{cases} u_j^{n+1} = u_j^n - \dfrac{a\Delta t}{4\Delta x}[(u_{j+1}^n - u_{j-1}^n) + (u_{j+1}^{n+1} - u_{j-1}^{n+1})] + \\ \qquad \dfrac{1}{2}Q_1(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}), \end{cases} \tag{3-58}$$

Q 在 0.1 – 0.3 之间变化.

二阶隐式格式为

$$\begin{cases} \overline{u_j^{n+1}} = u_j^n - \dfrac{a\Delta t}{4\Delta x}[(u_{j+1}^n - u_{j-1}^n) + (\overline{u_{j+1}^{n+1}} - \overline{u_{j-1}^{n+1}})] + \\ \qquad \dfrac{1}{2}Q_1(\overline{u_{j+1}^{n+1}} - 2\overline{u_j^{n+1}} + \overline{u_{j-1}^{n+1}}), \\ u_j^{n+1} = \overline{u_j^{n+1}} - \dfrac{1}{2}Q_1(u_{j+1}^n - 2u_j^n + u_{j-1}^n). \end{cases} \tag{3-59}$$

混合格式为

$$u_j^{n+1} = \left[\overline{u_j^{n+1}} - \frac{1}{2}Q_1(u_{j+1}^n - 2u_j^n + u_{j-1}^n)\right](1-\theta) + \theta\,\overline{u_j^{n+1}}. \tag{3-60}$$

对于非线性方程可以写为如下形式：

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}(f_{j+1}^n - f_{j-1}^n) - \frac{\Delta t}{4\Delta x}[a_{j+1}^n(u_{j+1}^{n+1} - u_{j+1}^n) - a_{j-1}^n(u_{j-1}^{n+1} - u_{j-1}^n)] + \frac{1}{2}Q_1(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) - \frac{1}{2}Q_1(1-\theta)(u_{j+1}^n - 2u_j^n + u_{j-1}^n). \tag{3-61}$$

该方法中 m、Q_1 是两个人为选用的参数，对计算结果有一定的影响，是一个不足之处.

3. NND 格式

该格式是一个无振荡无自由参数的耗散差分格式.在反扩散格式中，间断处附近是以损失精度为代价来消除振荡的，而且还有自由参数需要选用，对结果产生影响.本格式是在分析修正方程时发现有些格式使波速向下游传，而另一些格式向上游传，因此考虑在间断处附近将它们组合起，即使波动向间断处汇集，从而使间断处成为波的陷阱，以达到消除波动的目的，同时又不降低精度.这里首先分析迎风二阶格式：

$$\begin{cases} \dfrac{u_j^{n+1}-u_j^n}{\Delta t}+a\dfrac{3u_j^n-4u_{j-1}^n+u_{j-2}^n}{2\Delta x}=0, & a>0, \\ \dfrac{u_j^{n+1}-u_j^n}{\Delta t}+a\dfrac{-3u_j^n+4u_{j+1}^n-u_{j+2}^n}{2\Delta x}=0, & a<0. \end{cases} \tag{3-62}$$

$a>0$ 时的修正方程为

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=a\left(\frac{\Delta x^2}{3}\frac{\partial^3 u}{\partial x^3}-\frac{\Delta x^3}{4}\frac{\partial^4 u}{\partial x^4}+\cdots\right), \tag{3-63}$$

这里 $\nu_3=\dfrac{a\Delta x^2}{3}>0$ 使波传播速度增加，而使波向下游推移；$\nu_4=-\dfrac{a\Delta x^3}{4}<0$ 起衰减作用，从而使计算得以稳定.

$a<0$ 时修正方程为

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=a\left(\frac{\Delta x^2}{3}\frac{\partial^3 u}{\partial x^3}+\frac{\Delta x^3}{4}\frac{\partial^4 u}{\partial x^4}+\cdots\right), \tag{3-64}$$

可以看出 $\nu_4<0$ 仍有稳定作用，而 $\nu_3<0$ 使得波传播速度减小，因此波向上游推移. 下面再看中心格式：

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2\Delta x}=0. \tag{3-65}$$

它的修正方程为

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=-\frac{a\Delta x^3}{6}\frac{\partial^3 u}{\partial x^3}+\cdots. \tag{3-66}$$

可以看出，该格式和格式(3-53)组合可以减小波传播速度的变化，起到抑制波动的产生和传播，为此格式为

$$\begin{cases} \dfrac{u_j^{n+1}-u_j^n}{\Delta t}=-\dfrac{3f_j^{+n}-4f_{j-1}^{+n}+f_{j-2}^{+n}}{2\Delta x}-\dfrac{f_{j+1}^{-n}-f_{j-1}^{-n}}{2\Delta x}, & \text{间断上游}, \\ \dfrac{u_j^{n+1}-u_j^n}{\Delta t}=-\dfrac{-3f_j^{-n}+4f_{j+1}^{-n}-f_{j+2}^{-n}}{2\Delta x}-\dfrac{f_{j+1}^{+n}-f_{j-1}^{+n}}{2\Delta x}, & \text{间断下游}. \end{cases} \tag{3-67}$$

其中

$$\begin{cases} a=\dfrac{\mathrm{d}f}{\mathrm{d}u},\ a^+=\dfrac{a+|a|}{2},\ a^-=\dfrac{a-|a|}{2}, \\ f^+=a^+u,\ f^-=a^-u. \end{cases} \tag{3-68}$$

这里可以看出：在间断上游，a^+ 部分用二阶迎风格式，$\nu_3>0$，a^- 部分用中心格式，ν_3 也大于零. 这就使各种波都向下游推移；在间断下游，所用的格式正相反，使各种波都向上游推移. 两者结合正好达到消除振荡的目的. 若引入记号

$$\begin{cases} \Delta f_{j-\frac12}=f_j-f_{j-1},\Delta f_{j+\frac12}=f_{j+1}-f_j, \\ \Delta u_{j-\frac12}=u_j-u_{j-1},\Delta u_{j+\frac12}=u_{j+1}-u_j, \end{cases}$$

则间断上游 $|\Delta u_{j+\frac12}|>|\Delta u_{j-\frac12}|$，间断下游 $|\Delta u_{j+\frac12}|<|\Delta u_{j-\frac12}|$. 将上、下游公式统一起来，得 NND 格式为

$$\begin{cases}\dfrac{\partial u}{\partial t}=-\dfrac{1}{\Delta x}(\hat{f}_{j+\frac{1}{2}}^{n}-\hat{f}_{j-\frac{1}{2}}^{n}),\\ \hat{f}_{j+\frac{1}{2}}=f_{j+\frac{1}{2},L}^{n}+f_{j+\frac{1}{2},R}^{n},\\ f_{j+\frac{1}{2},L}^{n}=f_{j}^{+n}+\dfrac{1}{2}\min\bmod(\Delta f_{j-1/2}^{+n},\Delta f_{j+1/2}^{+n}),\\ f_{j+\frac{1}{2},R}^{n}=f_{j+1}^{-n}-\dfrac{1}{2}\min\bmod(\Delta f_{j+1/2}^{-n},\Delta f_{j+3/2}^{-n}).\end{cases} \tag{3-69}$$

$$\min\bmod(a,b)=\begin{cases}\operatorname{sgn}(a)\cdot\min\{|a|,|b|\}, & ab>0,\\ 0, & ab<0.\end{cases} \tag{3-70}$$

本格式由我国学者张涵信提出，他已证明，$\frac{\partial u}{\partial t}$ 用一阶差分时稳定条件为 $\frac{|a|\Delta t}{\Delta x}<\frac{2}{3}$，用二阶差分时稳定条件为 $\frac{|a|\Delta t}{\Delta x}<1$. 该格式计算结果表明，一般不出现波动(个别 $\frac{\partial^2 u}{\partial x^2}=0$ 处可能出现波动，但不大)，又不需选用参数具有二阶精度. 目前已被广泛用于气体动力学中的计算，并取得很大的成功.

4. TVD 格式

鉴别振荡大小的一个方法是确定总变差量 $TV(u)=\sum_j|u_{j+1}-u_j|$. 显然在振荡剧烈处，总变差量会是很大的，在平缓变化区总变差量将会变得比较小. 因此如果将总变差量不断减小，则振荡就会被抑制和衰减掉. 使总变差不断减小的差分格式就是 TVD 格式. 该格式是由 Harten 最早提出和发展的，后来人们进行了广泛的研究和发展，在气体动力学计算中成了最重要的计算格式. 关于该格式将在第 4 节中专门加以讨论.

以上的各种格式虽然是在对伯格方程的讨论中得到的，但它们都可以推广到方程组的情况中去，只要作以下的变动：

u、f 分别用 $\boldsymbol{U}$、$\boldsymbol{F}$ 来代替，后者均为向量列，在气体动力学中的表达形式已在第一章中详细列出了. 另外 a 用 $\boldsymbol{A}=\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}$ 代替，是一个矩阵，它可以作如下分解：

$$\boldsymbol{A}=\boldsymbol{R\Lambda L},\quad \boldsymbol{R}=\boldsymbol{L}^{-1},\quad \boldsymbol{L}=\boldsymbol{R}^{-1},$$

$$\boldsymbol{A}^{\pm}=\boldsymbol{R\Lambda}^{\pm}\boldsymbol{L},\quad \boldsymbol{\Lambda}=\operatorname{diag}(\lambda_i),$$

$$\boldsymbol{\Lambda}^{\pm}=\operatorname{diag}\left(\frac{\lambda_i\pm|\lambda_i|}{2}\right)=\operatorname{diag}(\lambda_i^{\pm}),\ \lambda_1,\lambda_2,\cdots,\lambda_n \text{ 为 } \boldsymbol{A} \text{ 的 } n \text{ 个特征}.$$

$$\boldsymbol{F}^{\pm}=\boldsymbol{A}^{\pm}\boldsymbol{U}.$$

而 $\min\bmod\{\boldsymbol{a},\boldsymbol{b}\}=\{\min\bmod(a_i,b_i)\}$. 这样以上讨论的各种格式都可以应用于方程组了.

3.3　迎风格式

迎风格式是由黎曼问题解得到启发而建立的，它又是以后格式的基础，所以本

节着重介结该格式的建立.

首先讨论伯格方程的黎曼问题的解.

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = 0, \\ u(x,0) = \begin{cases} u_-, & x < 0, \\ u_+, & x > 0. \end{cases} \end{cases} \tag{3-71}$$

其解为

(1) 当 $u_- > u_+$ 时

$$u(x,t) = \begin{cases} u_-, & x < st, \\ u_+, & x > st, \end{cases} \quad s = \frac{1}{2}(u_- + u_+). \tag{3-72}$$

(2) 当 $u_- < u_+$ 时

$$u(x,t) = \begin{cases} u_-, & x < u_- t, \\ \dfrac{x}{t}, & u_- t < x < u_+ t, \\ u_+, & x > u_+ t. \end{cases} \tag{3-73}$$

图 3-4 显示了解的图案. 当 $u_- > u_+$ 时,间断保持不变,以激波速度 s 向前推进;当 $u_- < u_+$ 时,生成一稀疏区,间断逐渐消失。

在 t 处,当 $L > st$ 时,

$$\begin{aligned} \frac{d}{dt}(2L\bar{u}) &= \frac{d}{dt}\int_{-L}^{L} u(x,t)dx \\ &= \int_{-L}^{L} \frac{\partial}{\partial t} u(x,t)dx = \int_{-L}^{L} -\frac{\partial f}{\partial x}dx = f_- - f_+, \end{aligned}$$

其中 $f_- = f(u_-)$ $f_+ = f(u_+)$. 另一方面

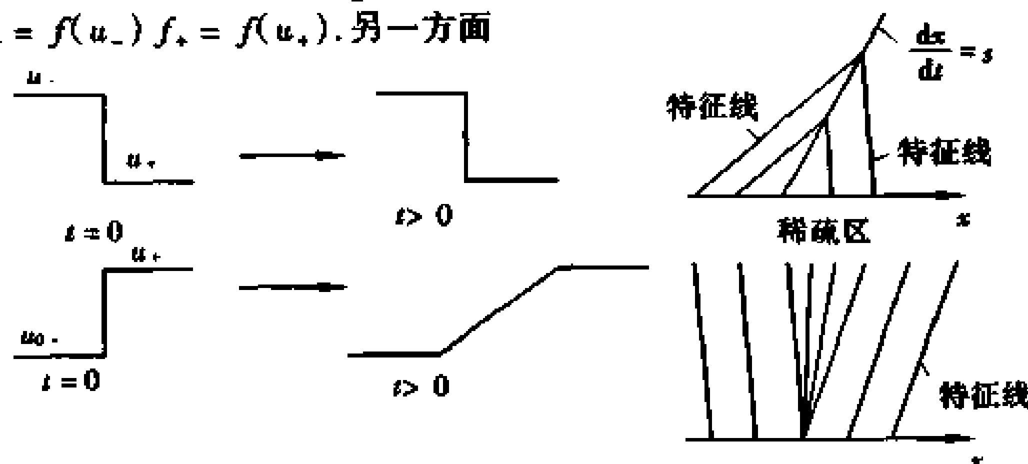


图 3-4

$$\frac{d}{dt}\int_{-L}^{L} u(x,t)dx = \frac{d}{dt}[(L+st)u_- + (L-st)u_-] = s(u_- - u_+).$$

二式比较得

$$s = \frac{f_- - f_+}{u_- - u_+}. \tag{3-74}$$

保持间断的条件为

$$\frac{f-f_-}{u-u_-} > s > \frac{f-f_+}{u-u_+}. \tag{3-75}$$

该结果推广到线性方程组

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x} = 0, \tag{3-76}$$

其中 $\boldsymbol{A}$ 为常数矩阵.上式改写为守恒型

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{AU}}{\partial x} = 0. \tag{3-77}$$

由于方程是双曲型的,故有

$$\boldsymbol{A} = \boldsymbol{S\Lambda S}^{-1}, \tag{3-78}$$

其中 $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$.引入

$$\boldsymbol{V} = \boldsymbol{S}^{-1}\boldsymbol{U}, \tag{3-79}$$

(3-77) 式可改定为

$$\frac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{\Lambda}\frac{\partial \boldsymbol{V}}{\partial x} = 0 \tag{3-80}$$

或

$$\frac{\partial v_i}{\partial t} + \lambda_i\frac{\partial v_i}{\partial x} = 0 \quad (i = 1,2,\cdots,n). \tag{3-81}$$

初值为

$$v_i(x,0) = \begin{cases} v_{i-}, & x < 0, \\ v_{i+}, & x > 0, \end{cases} \tag{3-82}$$

解为

$$v_i(x,t) = v_i(x-\lambda_i t,0), \tag{3-83}$$

最后得

$$\boldsymbol{U}(x,t) = \boldsymbol{SV}(x,t), \tag{3-84}$$

其中 $\boldsymbol{S}$ 中各列为 $\boldsymbol{A}$ 矩阵的右特征向量 $\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_n$, 即

$$\boldsymbol{S} = (\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_n), \tag{3-85}$$

因此

$$\boldsymbol{U}(x,t) = \sum_{i=1}^{n} v_i(x-\lambda_i t,0)\boldsymbol{r}_i. \tag{3-86}$$

这里的解 $\boldsymbol{U}(x,t)$ 不只和初值有关,还和区域有关.为清楚起见,设有三个特征(即 $n=3$),则解可如图 3-5 所示.

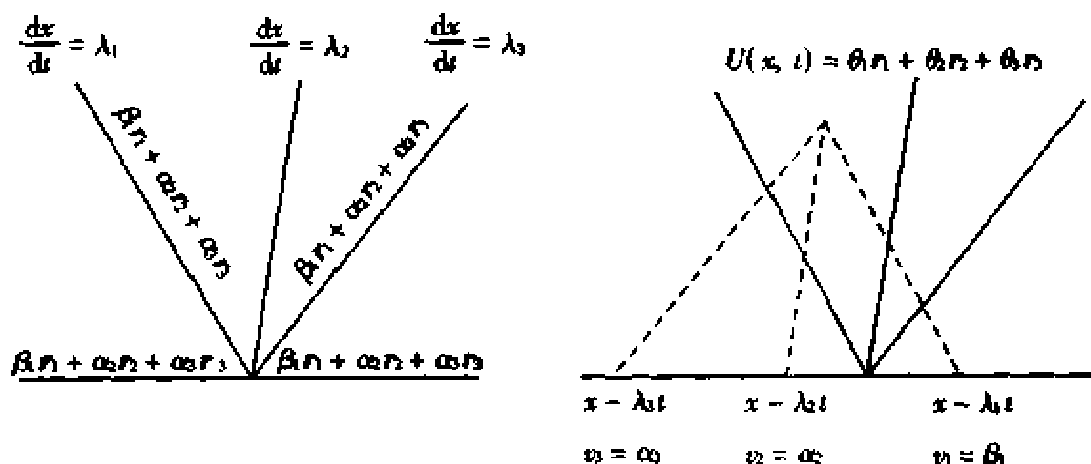


图 3-5

图3-5为 $x=0$ 处发出的三条特征线，将 $x-t$ 平面分解成四个区，在每个区域内的解 $U(x,t)=\theta_1 r_1+\theta_2 r_2+\theta_3 r_3$. 其中 θ_i 的确定是这样的：通过 (x,t) 点作三条特征线，设第 i 条特征线与 $t=0$ 交于 $x<0$ 处时，$\theta_i=\alpha_i$；交于 $x>0$ 处时 $\theta_i=\beta_i$. α_i、β_i 即所在处的 v_i 的初值. 当初值存在间断时，间断将沿特征线传播，即穿过 λ_i 特征时的间断

$$\begin{cases}[U]_i=(\beta_i-\alpha_i)r_i \quad (i=1,2,\cdots,n),\\ [F]_i=A[U]_i=(\beta_i-\alpha_i)Ar_i\\ \qquad =(\beta_i-\alpha_i)\lambda_i r_i=\lambda_i[U]_i\end{cases}$$

还可写作

$$U(x,t)=U_-+\sum_{\lambda_i<x/t}(\beta_i-\alpha_i)r_i=U_+-\sum_{\lambda_i>x/t}(\beta_i-\alpha_i)r_i,$$

$$U_+-U_-=\sum_{i=1}^{n}(\beta_i-\alpha_i)r_i,$$

其中 $\sum\limits_{\lambda_i>x/t}$ 表示 $\lambda_i>\frac{x}{t}$ 的 i 求和，$\sum\limits_{\lambda_i<x/t}$ 表示 $\lambda_i<\frac{x}{t}$ 的 i 求和.

这里 U 的 n 个分量 $u_1,u_2,\cdots,u_n$ 构成 n 维相空间. 当 U_+-V_- 与 A 的 n 个特征向量之一 r_i 平行，即 $U_+-U_-/\!/ r_i$，则该间断将在特征线上保持，否则将分解为 n 个间断. 这种保持单一间断的情况叫作 **Hugoniot 轨迹**. Hugoniot 轨迹是一条曲线，通过相空间内任一点 $U(u_1,u_2,\cdots,u_n)$ 可以作出 n 条直线，直线方向分别与 $r_1,r_2,\cdots,r_n$ 平行. 如果 U_+ 和 U_- 不在同一族线上，则是一个一般的情况，它们将分解为 n 个间断，并沿 n 个不同的方向传播. 方程的解 $U(x,t)$ 在不同区域内的值和上面讨论的 $n=3$ 的情况相类似，即

$$U(x,t)=\begin{cases}\alpha_1 r_1+\cdots+\alpha_n r_n, & \dfrac{dx}{dt}<\lambda_1,\\ \alpha_1 r_1+\cdots+\alpha_i r_i+\beta_{i+1}r_{i+1}+\cdots+\beta_n r_n, & \lambda_i<\dfrac{dx}{dt}<\lambda_{i+1}, i=1,2,\cdots,n-1,\\ \beta_1 r_1+\cdots+\beta_n r_n, & \dfrac{dx}{dt}>\lambda_n.\end{cases}$$

(3-87)

以上讨论的是线性方程组的情况，下面讨论非线性方程组的情况. 方程为

$$\frac{\partial U}{\partial t}+\frac{\partial F(U)}{\partial x}=0\Leftrightarrow\frac{\partial U}{\partial t}+A(U)\frac{\partial U}{\partial x}=0 \tag{3-88}$$

其中矩阵 A 的 n 个特征值为

$$\lambda_1<\lambda_2<\cdots<\lambda_n,$$

对应的特征向量 $r_1,r_2,\cdots,r_n$ 线性无关，并且归一化，即 $|r_i|=1(i=1,2,\cdots,n)$，且它们都是 U 的函数. 在线性问题中间断以特征速度 (λ_i) 传播. 但在非线性问题中间断应满足的条件是

$$[F]=s[U], \tag{3-89}$$

其中 s 为激波的速度. 若激波一侧的 U 量固定不变，满足(3-98)式条件的另一侧 U

和 s 可以有许多个，记前者为 U_+，后者为 U_-. 这里 U_- 和 s 是未知的，共有 $n+1$ 个量. 由于解有无限个，故可将解视为含有一参数，记作 ζ，于是这个解可记作

$$\begin{cases} U_{i-}(s, U_+) = U_+ + \zeta r_i, \\ s_i(\zeta, U_+) = \lambda_i(U_+) \quad (i = 1, 2, \cdots, n). \end{cases} \tag{3-90}$$

对于线性问题，r_i 是常向量，故 U_- 落在过 U_+ 点的相空间内的 n 条直线上. 对于非线性问题，(3-90) 式表示 n 条曲线，在这些曲线上

$$F[U_{i-}(\zeta, U_+)] - F(U_+) = s_i(\zeta, U+)[U_{i-}(\zeta, U_+) - U_+], \quad i = 1, 2, \cdots, n. \tag{3-91}$$

方程两侧对 ζ 求导并令 $\zeta = 0$，则得

$$F'[U_{i-}(0, U_+)]U'_{i-}(0, U_+) = s_i(0, U_+)U'_{i-}(0, U_+).$$

由(3-90) 式知 $U_{i-}(0, U_+) = U_+, s_i(0, U_+) = \lambda_i(U_+)$，故上式可改写为

$$F'(U)_- \cdot U'_{i-}(0, U_+) = \lambda_i(U_+) \cdot U'_{i-}(0, U_+).$$

由 $\lambda_i(U_+)$ 就是 $F'(U_+)$ 的特征向量，故 $U'_{i-}(0, U_+)$ 必平行于 r_i，也就是说过 U_+ 点的相空间内的曲线在 U_+ 处的切线与 U_+ 处的 r_i 方向平行. 这些曲线叫 **Hugoniot 曲线**. 所有这些曲线上的点集合叫做 U_+ 点的 Hugoniot 轨迹. 如果 U_- 落在第 i 条曲线上，就与 U_+ 构成第 i 族波. 为构造黎曼问题的解，过 U_L 和 U_R 作 Hugoniot 曲线，取满足熵条件的交点. 应当指出黎曼解可以有间断，也可以无间断. 无间断的解应是自相似解，即解为 x/t 的函数. 利用以上方法可以确定非线性双曲型方程的一般形式的黎曼解. 关于一维气体动力学方程的黎曼解已在第 1 章中介绍过了.

迎风格式的基本思想是：在各离散点上的值可以看作该点邻域内的平均值，故函数值分布是一台阶函数，在每相邻两个单元的交接处就构成了一个黎曼问题，用上面方法得到的黎曼解可以得到 Δt 时间间隔后的物理量的新的分布，只要气体速度 u 和当地声速 c 满足条件

$$(|u| + c)_{\max_{\forall j}} \frac{\Delta t}{\Delta x} \leqslant 1. \tag{3-92}$$

不同交接面上的黎曼解是互不相干的，这也就是 Godunov 格式的稳定条件. 将所得的黎曼解在 $t + \Delta t$ 处的分布值在离散点的邻域内再次平均，就可以得到离散点上新的物理量的值. 这一过程就构成了 Godunov 格式. 考虑到平均过程十分繁琐，可以由单元两侧的通量差得到单元内量的变化. 由于黎曼解的复杂性，具体建立过程中还需用线化方法作一些简化，得

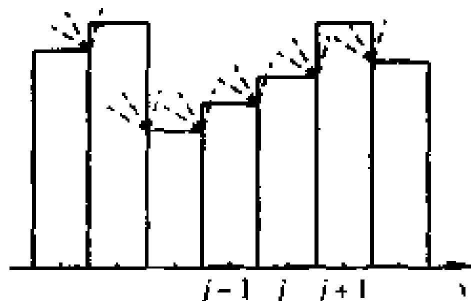


图 3-6

$$\rho_j^{n+1} = \rho_j^n - \frac{\Delta t}{\Delta x}\{(\rho u)_{j+\frac{1}{2}} - (\rho u)_{j-\frac{1}{2}}\},$$

$$(\rho u)_j^{n+1} = (\rho u)_j^n - \frac{\Delta t}{\Delta x}\{(p+\rho u^2)_{j+\frac{1}{2}} - (p+\rho u^2)_{j-\frac{1}{2}}\},$$

$$[\rho(e+\frac{u^2}{2})]_j^{n+1} = [\rho(e+\frac{u^2}{2})]_j^n - \frac{\Delta t}{\Delta x}\{[\rho u(e+\frac{u^2}{2}+\frac{p}{\rho})]_{j+\frac{1}{2}} - [\rho u(e+\frac{u^2}{2}+\frac{p}{\rho})]_{j-\frac{1}{2}}\},$$

其中 $j\pm\frac{1}{2}$ 处的值可取黎曼解在 $j\pm\frac{1}{2}$、$n+\frac{1}{2}$ 处的值.这个值可以这样计算：

记 $p^* = p_{j+\frac{1}{2}}, u^* = u_{j+\frac{1}{2}}$,

$$\begin{cases} p^* = \dfrac{b_{j+\frac{1}{2}}p_j + a_{j+\frac{1}{2}}p_{j+1} + a_{j+\frac{1}{2}}b_{j+\frac{1}{2}}(u_j - u_{j+1})}{a_{j+\frac{1}{2}}b_{j+\frac{1}{2}}}, \\ u^* = \dfrac{a_{j+\frac{1}{2}}u_{j+\frac{1}{2}} + b_{j+\frac{1}{2}}u_{j+1} + p_j - p_{j+1}}{a_{j+\frac{1}{2}} + b_{j+\frac{1}{2}}}, \end{cases} \tag{3-93}$$

其中

$$a_{j+\frac{1}{2}} = \begin{cases} \sqrt{\dfrac{1}{2}[(\gamma+1)p^* + (\gamma-1)p_j]\rho_j}, & p^* \geqslant p_j, \\ \dfrac{\gamma-1}{2\gamma}\sqrt{\gamma p_j\rho_j}\,\dfrac{1-p^*/p_j}{1-(p^*/p_j)^{\frac{\gamma-1}{2r}}}, & p^* < p_j, \end{cases} \tag{3-94}$$

$$b_{j+\frac{1}{2}} = \begin{cases} \sqrt{\dfrac{1}{2}[(\gamma+1)p^* + (\gamma-1)p_{j+1}]\rho_{j+1}}, & p^* \geqslant p_{j+1}, \\ \dfrac{\gamma-1}{2r}\sqrt{rp_{j+1}\rho_{j+1}}\,\dfrac{1-p^*/p_{j+1}}{1-(p^*/p_{j+1})^{\frac{\gamma-1}{2r}}}, & p^* < p_{j+1}. \end{cases} \tag{3-95}$$

有了 u^*、p^*,可以计算 $\rho^* = \rho_{j+\frac{1}{2}}$ 为

$$\rho^* = \begin{cases} \rho_j(p^*/p_j)^{\frac{1}{\gamma}}, & p_j \geqslant p_{j+1}, \\ \rho_{j+1}(p^*/p_{j+1})^{\frac{1}{\gamma}}, & p_j < p_{j+1}. \end{cases} \tag{3-96}$$

不难看出,确定 p^*、u^* 是一个求解非线性方程组的问题,计算比较困难,若相邻两点之间的物理量相差不大,则可以作弱波处理,这时近似地有

$$\begin{cases} a_{j+\frac{1}{2}} = b_{j+\frac{1}{2}} = \sqrt{\dfrac{r}{4}(p_j + p_{j+1})(\rho_j + \rho_{j+1})}, \\ p^* = \dfrac{p_j + p_{j+1}}{2} + a_{j+\frac{1}{2}}\dfrac{u_j - u_{j+1}}{2}, \\ u^* = \dfrac{u_j + u_{j+1}}{2} + \dfrac{p_j - p_{j+1}}{2a_{j+\frac{1}{2}}}. \end{cases} \tag{3-97}$$

有了 p^*、u_*、ρ^*,可以确定 $j+\frac{1}{2}$ 处的 e;$j-\frac{1}{2}$ 处的 p^*、u^*、ρ^*、e 也用同样方法确定.这就给出了 Godunov 格式求解的全过程.不难看出,这一格式是一阶精度的.

有较高的耗散.为了提高精度,在离散点的邻域内物理量不看作常数,而看作线性变化的.这样做使交接面上的间断变弱,精确度可以提高.Godunov 格式还有一些其它改进型的表达式.这里不一一详细介绍.

Godwnov 格式由于其精度低,使间断抹平,在气体动力学计算中已不再广泛应用,但它的构造(从黎曼解出发)是富有启发性的,在以后构造其它格式时极有参考价值.因此它的影响不可低估.

3.4　TVD 格式

TVD(total variation diminishing) 格式是 Harten 最早提出和发展的,它的基本思想是通过限制和减少解的总变差来达到抑制乃至消除非物理振荡的目的.设差分格式一般地表示为

$$u_j^{n+1} = L_h u_j^n, \tag{3-98}$$

当

$$TV(u^{n+1}) < TV(u^n) \tag{3-99}$$

时格式(3-98)称之为 **TVD 格式**,其中

$$TV(u^n) = \sum_j |u_{j+1}^n - u_j^n|. \tag{3-100}$$

不难证明 TVD 格式必定是保单格式.用反证法易于证明这一点.设

$$u_L \leqslant \cdots \leqslant u_{j-1}^n \leqslant u_j^n \leqslant u_{j+1}^n \leqslant \cdots \leqslant u_R,$$

故

$$TV(u^n) = u_R - u_L.$$

在第 $n+1$ 层上不保单,即

$$u_L \leqslant \cdots \leqslant u_{j-1}^{n+1},\quad u_{j-1}^{n+1} > u_j^{n+1}, u_j^{n+1} \leqslant \cdots \leqslant u_R,$$

则

$$\begin{aligned} TV(u^{n+1}) &= \sum_{k\leqslant j-2} |u_{k+1}^{n+1} - u_k^{n+1}| + \sum_{k\geqslant j} |u_{k+1}^{n+1} - u_k^{n+1}| + |u_j^{n+1} - u_{j-1}^{n+1}| \\ &= u_R - u_L + 2(u_j^{n+1} - u_{j+1}^{n+1}) > u_R - u_L = TV(u^n). \end{aligned}$$

可见总变差就不再减少.这表明 TVD 格式必定是保单格式.反之亦然.

TVD 格式的一个充分条件是,若格式写为

$$u_j^{n+1} = u_j^n + C_{j+\frac{1}{2}}^n(u_{j+1}^n - u_j^n) - D_{j-\frac{1}{2}}^n(u_j^n - u_{j-1}^n), \tag{3-101}$$

其中

$$C_{j+\frac{1}{2}}^n, D_{j+\frac{1}{2}}^n \geqslant 0, C_{j+\frac{1}{2}} D_{j+\frac{1}{2}} \leqslant 1,\quad \forall (j,n),$$

则该格式是 TVD 的.

事实上

$$\begin{aligned} TV(u^{n+1}) &= \sum_j |u_{j+1}^{n+1} - u_j^{n+1}| \\ &\leqslant \sum_j \{|u_{j+1}^n - u_j^n|(1 - C_{j+\frac{1}{2}}^n - D_{j+\frac{1}{2}}^n) + C_{j+\frac{3}{2}}^n |u_{j+2}^n - u_{j+1}^n| + \\ &\quad D_{j-\frac{1}{2}}^n |u_j^n - u_{j-1}^n|\} \\ &= \sum_j \{|u_{j+1}^n - u_j^n|(1 - C_{j+\frac{1}{2}}^n - D_{j+\frac{1}{2}}^n) + C_{j+\frac{1}{2}}^n |u_{j+1}^n - u_j^n| + \\ &\quad D_{j+\frac{1}{2}}^n |u_{j+1}^n - u_j^n|\} \end{aligned}$$

$$= TV(u^n).$$

为提高稳定性可采用格式如下：

$$Yu_j^{n+1} = Zu_j^n \tag{3-102}$$

其中 Z 为 TVD 的，而 Y 为 TVI(总变差增加) 的. 这样得到的格式

$$u_j^{n+1} = Y^{-1}Zu_j^n$$

是 TVD 的.

在讨论 TVD 格式构造之前，首先讨论一般守恒型方程差分格式的一些特性.

单变量守恒型方程(2-43) 的差分格式的一般形式为

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x} = 0, \tag{3-103}$$

其中 $\hat{f}$ 为数值通量，即

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}(u_{j-k+1}, u_{j-k+2}, \cdots, u_j, \cdots, u_{j+k}), \tag{3-104}$$

它是一个连续函数，且满足相容条件

$$\hat{f}(u, u, \cdots, u) = f(u). \tag{3-105}$$

如在弗里德里希 - 拉克斯格式中

$$\hat{f}_{j+\frac{1}{2}} = \frac{1}{2}[f(u_{j+1}) + f(u_j) - \frac{\Delta x}{\Delta t}(u_{j+1} - u_j)];$$

在拉克斯 - 温德罗夫格式中

$$\hat{f}_{j+\frac{1}{2}} = \frac{1}{2}[f(u_{j+1}) + f(u_j) - \frac{\Delta x}{\Delta t}a_{j+\frac{1}{2}}(u_{j+1}^n - u_j^n)],$$

其中

$$a = \frac{\partial f}{\partial u}, \quad a_{j+\frac{1}{2}} = a\left(\frac{u_j + u_{j+1}}{2}\right);$$

在 Mac Cormack 格式中

$$\hat{f}_{j+\frac{1}{2}} = \frac{1}{2}[f(u_{j+1}) + f(u_j^n - \frac{\Delta t}{\Delta x}(f(u_{j+1}^n) - f(u_j^n))];$$

在 Godunov 格式中

$$\hat{f}_{j+\frac{1}{2}} = f(V(0, u_{j+1}, u_j)),$$

其中 $V(\frac{x}{t}, u_{j+1}, u_j)$ 为 $j + \frac{1}{2}$ 处黎曼解的值.

不难看出格式(3-103) 具有守恒性.

引入记号

$$\begin{cases} G(u_{j-k}, u_{j-k+1}, \cdots, u_{j+k}) = u_j^n - \frac{\Delta t}{\Delta x}[\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}], \\ \frac{\partial G}{\partial u_{j-l}} = G_{,l}, \quad \frac{\partial \hat{f}_{j\pm\frac{1}{2}}}{\partial u_{j-l}} = \hat{f}_{j\pm\frac{1}{2},l}, \end{cases}$$

经繁琐的运算可以得到格式(3-113) 的修正方程为

$$\frac{\partial u}{\partial t}+\frac{\partial f}{\partial x}=\frac{\Delta t}{2}\frac{\partial}{\partial x}\left[\beta\left(u,\frac{\Delta t}{\Delta x}\right)\frac{\partial u}{\partial x}\right]+\frac{\Delta t^2}{6}\frac{\partial}{\partial x}\left[\gamma\left(u,\frac{\Delta t}{\Delta x}\right)\frac{\partial^2 u}{\partial x^2}+\delta\left(u,\frac{\Delta t}{\Delta x}\right)\frac{\partial u}{\partial x}\frac{\partial u}{\partial x}\right]+\cdots,\tag{3-106}$$

其中

$$\begin{cases}\beta\left(u,\dfrac{\Delta t}{\Delta x}\right)=\left(\dfrac{\Delta x}{\Delta t}\right)^2\displaystyle\sum_{r=j-k}^{j+k}(r-j)^2G_{,r-j}-\left(\dfrac{\partial f}{\partial u}\right)^2,\\ \gamma\left(u,\dfrac{\Delta t}{\Delta x}\right)=\left(\dfrac{\Delta x}{\Delta t}\right)^3\displaystyle\sum_{r=j-k}^{j+k}(r-j)^2G_{,r-j}+\left(\dfrac{\partial f}{\partial u}\right)^2,\\ \delta\left(u,\dfrac{\Delta x}{\Delta t}\right)=\left(\dfrac{\Delta x}{\Delta t}\right)^3\displaystyle\sum_{s,r=j-k}^{j+k}\dfrac{(r-j)+(s-j)}{4}[2(r-j)(s-j)-\\ \qquad (r-s)^2]G_{r-j,s-j}+3\left(\dfrac{\partial f}{\partial u}\right)^2\dfrac{\partial^2 f}{\partial u^2}.\end{cases}\tag{3-107}$$

利用这些公式可以用来分析各种格式的精度.

另外对应守恒方程的熵守恒方程为

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=0,\tag{3-108}$$

在间断处熵条件为

$$\oint U(u)\mathrm{d}x-F(u)\mathrm{d}x\geqslant 0,\tag{3-109}$$

或导致

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}\leqslant 0.\tag{3-110}$$

相应的弱解应当满足的熵条件为

$$\frac{U(u_j^{n+1})-U(u_j^n)}{\Delta t}+\frac{\widehat{F}_{j+\frac{1}{2}}-\widehat{F}_{j-\frac{1}{2}}}{\Delta x}\leqslant 0,\tag{3-111}$$

其中数值熵通量为

$$\widehat{F}_{j+\frac{1}{2}}=\widehat{F}(u_{j-k+1}^n,\cdots,u_{j+k}^n),\tag{3-112}$$

并且满足相容条件

$$\widehat{F}(u,u,\cdots,u)=F(u).\tag{3-113}$$

拉克斯证明了以下定理:

定理　若 Δx、$\Delta t\to 0$ 时满足熵条件(3-111)的守恒格式(3-103)的差分解 $u_n(x,t)$ 一致有界,而且几乎处处收敛于分片连续可微函数 $u(x,t)$,则此极限函数 $u(x,t)$ 是守恒方程(2-43)的可允许弱解.

应当指出,该定理只对单变量方程作出了证明,对于多变量方程组的证明尚需研究.另外,有时人们常用半离散化的格式

$$\frac{\partial u_j}{\partial t}+\frac{\widehat{f}_{j+\frac{1}{2}}-\widehat{f}_{j-\frac{1}{2}}}{\Delta x}=0, \tag{3-114}$$

则相应的熵条件为

$$\frac{\partial U(u_j)}{\partial t}+\frac{\widehat{F}_{j+\frac{1}{2}}-\widehat{F}_{j-\frac{1}{2}}}{\Delta x}\leqslant 0. \tag{3-115}$$

这些条件可以用来判别格式是否满足这样的离散熵条件.

有了以上的讨论,可以进一步讨论 TVD 格式的构造. 由于前面提到保单格式是 TVD 格式,所以构造 TVD 格式的一个方法是从保单格式入手.

若记

$$u_j^{n+1}=\mathrm{L}_h u_j^n=H(u_{j-k}^n,u_{j-k+1}^n,\cdots,u_{j+k}^n) \tag{3-116}$$

或记

$$u_j^{n+1}=\mathrm{L}_h u_j^n=\sum_{l=-k}^{k}C_l u_{j+l}^n, \tag{3-117}$$

相比较可得

$$C_l=\frac{\partial H}{\partial u_{j+l}^n}. \tag{3-118}$$

当$\dfrac{\partial H}{\partial u_{j+l}^n}\geqslant 0$时,上述格式就是保单格式,也称单调格式. 可以证明该格式满足离散熵条件.

1. TVD 的一阶格式

一般守恒型格式具有如下形式:

$$\begin{cases}u_j^{n+1}=u_j^n-\dfrac{\Delta t}{\Delta x}(\widehat{f}_{j+\frac{1}{2}}-\widehat{f}_{j-\frac{1}{2}}),\\ \widehat{f}_{j+\frac{1}{2}}=\dfrac{1}{2}\left[f(u_j^n)+f(u_{j+1}^n)-\dfrac{\Delta x}{\Delta t}Q\left(\dfrac{\overline{a}_{j+\frac{1}{2}}^n\Delta t}{\Delta x}\right)(u_{j+1}^n-u_j^n)\right],\end{cases} \tag{3-119}$$

其中

$$\overline{a}_{j+\frac{1}{2}}^n=\begin{cases}\dfrac{f(u_{j+1}^n)-f(u_j^n)}{u_{j+1}^n-u_j^n}, & u_{j+1}^n\neq u_j^n\\ a(u_j^n), & a=\dfrac{\partial f}{\partial u},u_{j+1}^n=u_j^n.\end{cases}$$

这里 Q 是人为选用的粘性系数,除保证格式相容性外,还要求保证 TVD 性质,即要求

$$|\eta|\leqslant Q(\eta)\leqslant 1\quad(0\leqslant|\eta|\leqslant\mu<1,\mu\text{ 为一小于 1 的确定值}).$$

事实上(3-119) 式可改写为

$$u_j^{n+1}=u_j^n+\frac{1}{2}\left[Q\left(\frac{\overline{a}_{j+\frac{1}{2}}^n\Delta t}{\Delta x}\right)-\frac{\Delta t}{\Delta x}\overline{a}_{j+\frac{1}{2}}^n\right](u_{j+1}^n-u_j^n)-$$
$$\frac{1}{2}\left[Q\left(\frac{\overline{a}_{j-\frac{1}{2}}^n\Delta t}{\Delta x}\right)-\frac{\Delta t}{\Delta x}\overline{a}_{j-\frac{1}{2}}^n\right](u_j^n-u_{j-1}^n),$$

不难看出，当$\frac{|a|\Delta t}{\Delta x} \leqslant \mu < 1$时，上式满足TVD格式充分条件(3-101).这也是该格式的稳定条件. $Q(\eta)$的选用可有三种：

$Q(\eta) = |\eta|$，(Roe格式)；

$$Q(\eta) = \begin{cases} |\eta|, & |\eta| > \varepsilon, \\ \varepsilon, & |\eta| \leqslant \varepsilon; \end{cases}$$

$$Q(\eta) = \begin{cases} |\eta|, & |\eta| > \varepsilon, \\ \dfrac{\eta^2 + \varepsilon^2}{2\varepsilon}, & |\eta| \leqslant \varepsilon. \end{cases}$$

分析表明，当$\varepsilon = 0$时，对上述第一种选择，格式不满足熵条件，可能产生非物理解.为此用第二、三种选择.又考虑到选第一种$Q(\eta)$，格式具有最小的耗散，所以ε应当比较小，一般选$\varepsilon = 0.025—0.125$.

2. TVD的二阶格式

大家知道，一阶迎风格式是一阶格式，L-W是二阶格式，但不是TVD格式，分别写出它们如下：

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left[\frac{a+|a|}{2}(u_j^n - u_{j-1}^n)\frac{a-|a|}{2}(u_{j+1}^n - u_j^n)\right],$$

$$u_j^{n+1} = u_j^n - \frac{\sigma+|\sigma|}{2}(u_j^n - u_{j-1}^n) - \frac{\sigma-|\sigma|}{2}(u_{j+1}^n - u_j^n) - \frac{1-|\sigma|}{2}\sigma(u_{j+1}^n - 2u_j^n + u_{j-1}^n).$$

比较可知，在$|\sigma| = \left|\frac{a\Delta t}{\Delta x}\right| < 1$时，L-W格式因附加了一项而使耗散减小，这就是前面提到过的反扩散法.为使方程具有TVD性质，又有二阶精度，可以改进这一附加项，即使

$$u_j^{n+1} = u_j^n - \frac{\sigma+|\sigma|}{2}(u_j^n - u_{j-1}^n) - \frac{\sigma-|\sigma|}{2}(u_{j+1}^n - u_j^n) - \frac{1-|\sigma|}{2}\sigma[\varphi(r_j^+)(u_{j+1}^n - u_j^n) - \varphi(r_j^-)(u_j^n - u_{j-1}^n)], \tag{3-120}$$

其中

$$r_j^+ = \frac{u_{j+2}^n - u_{j+1}^n}{u_{j+1}^n - u_j^n},$$

$$r_j^- = \frac{u_j^n - u_{j-1}^n}{u_{j+1}^n - u_j^n},$$

而$\varphi(r)$应在图3-7的范围内.

不同的限制器给出了不同的格式，可见图3-7所示.

建立二阶格式还可以用通量修正法，其思想是改变原守恒方程的通量项，即

$$\frac{\partial u}{\partial t} + \frac{\partial f^M}{\partial x} = 0, \quad f^M = f + \frac{\Delta t}{\Delta x}g(x). \tag{3-121}$$

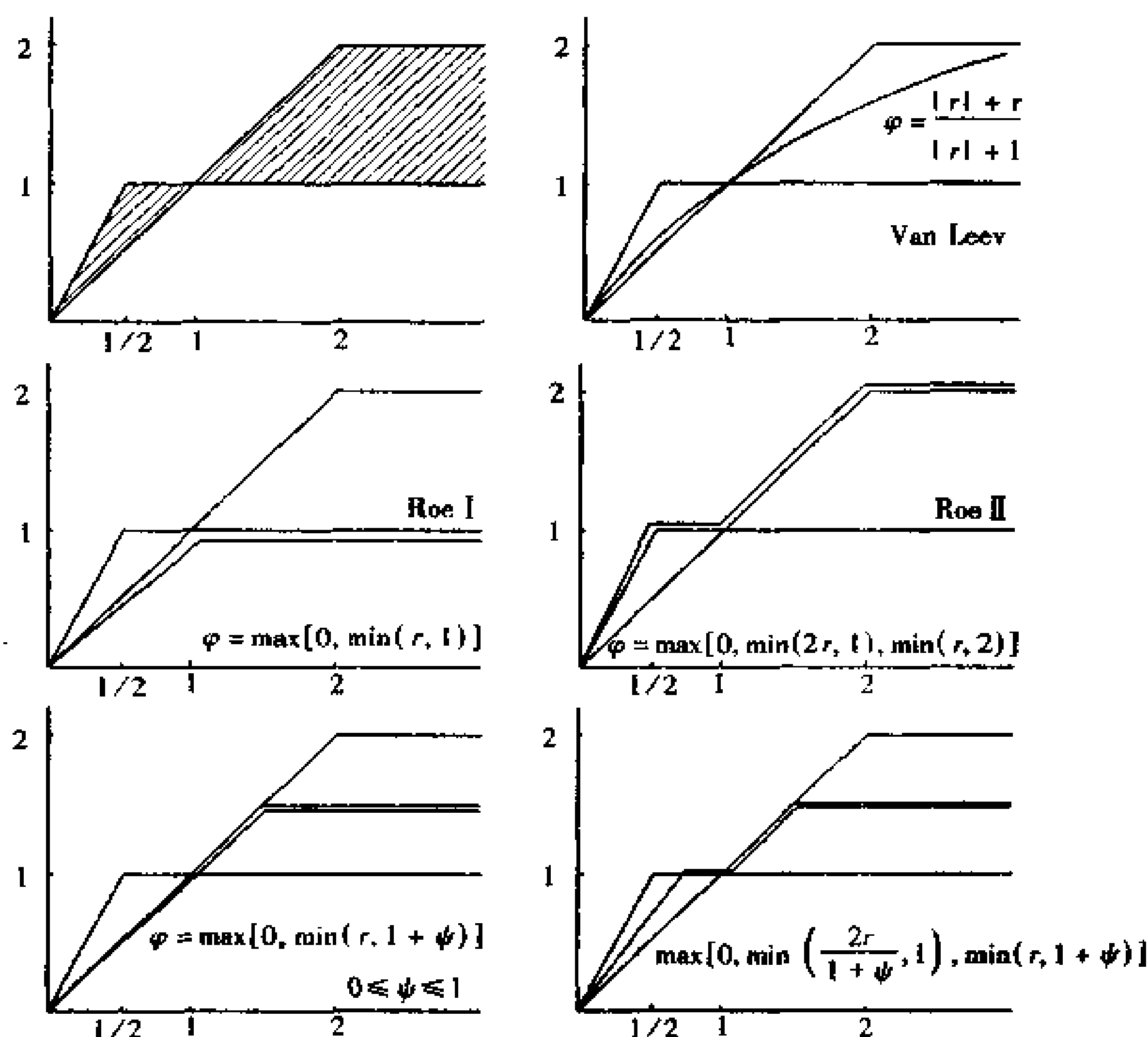


图 3-7

适当选用 g,使得采用以上各种一阶格式时得到的方程与原方程相容并有二阶精度.这是 Harten 提出的设想,具体作法如下:

$$
\begin{cases}
u_j^{n+1} = u_j^n - \dfrac{\Delta t}{\Delta x}\left(\hat{f}_{j+\frac{1}{2}}^M - \hat{f}_{j-\frac{1}{2}}^M\right), \\
\hat{f}_{j+\frac{1}{2}}^M = \dfrac{1}{2}\Big(f(u_j^n) + \dfrac{\Delta x}{\Delta t}g(u_j^n) + f(u_{j+1}^n) + \dfrac{\Delta x}{\Delta t}g(u_{j+1}^n) - \\
\qquad \dfrac{\Delta x}{\Delta t}Q\left(\dfrac{\Delta t}{\Delta x}\bar{a}_{j+\frac{1}{2}}^n + \gamma_{j+\frac{1}{2}}\right)(u_{j+1}^n - u_j^n), \\
\gamma_{j+\frac{1}{2}} = \begin{cases} \dfrac{g(u_{j+1}^n) - g(u_j^n)}{u_{j+1}^n - u_j^n}, & u_{j+1}^n \neq u_j^n, \\ 0, & u_{j+1}^n = u_j^n, \end{cases} \\
g_j = \min\operatorname{mod}(\tilde{g}_{j+\frac{1}{2}}, \tilde{g}_{j-\frac{1}{2}}), \\
\tilde{g}_{j+\frac{1}{2}} = \dfrac{1}{2}\left[Q\left(\dfrac{\Delta t}{\Delta x}\bar{a}_{j+\frac{1}{2}}\right) - \left(\dfrac{\Delta t}{\Delta x}\bar{a}_{j+\frac{1}{2}}\right)^2\right](u_{j+1}^n - u_j^n)).
\end{cases}
\tag{3-122}
$$

其中 $\bar{a}$ 的定义同前，$Q(\eta)$ 的作法也同前.该方法已由 Harten 证明具有二阶精度，是 TVD 的，而且满足熵条件.该方法在 u 的极值处会退化为一阶精度.

在一阶精度格式中，$x_{j+\frac{1}{2}}$ 处的左、右函数值用 u_j 和 u_{j+1}，为了提高精度，可以设 $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ 之间 u 的变化是线性的，于是 $x_{j+\frac{1}{2}}$ 两侧的值记作 $u_{j+\frac{1}{2}R}$ 和 $u_{j+\frac{1}{2}L}$，即

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\hat{f}_{j+\frac{1}{2}}^n - \hat{f}_{j-\frac{1}{2}}^n), \tag{3-123}$$

其中 $\hat{f}_{j+\frac{1}{2}} = f^+(\hat{u}_{j+\frac{1}{2},L}) + f^-(\hat{u}_{j+\frac{1}{2},R})$,

$$\hat{u}_{j+\frac{1}{2},L} = u_j + \frac{1}{4}[(1-k)\bar{\Delta}_- + (1+k)\bar{\Delta}_+]_j,$$

$$\hat{u}_{j+\frac{1}{2},R} = u_{j+1} - \frac{1}{4}[(1-k)\bar{\Delta}_+ + (1+k)\bar{\Delta}_-]_j,$$

$$\bar{\Delta}_+ = \max[0, \min(\Delta_+ \operatorname{sgn}\Delta_-, b\Delta_- \operatorname{sgn}\Delta_+)]\operatorname{sgn}\Delta_- = \min\operatorname{mod}(\Delta_+, b\Delta_-),$$

$$\bar{\Delta}_- = \max[0, \min(\Delta_- \operatorname{sgn}\Delta_+, b\Delta_+ \operatorname{sgn}\Delta_+)]\operatorname{sgn}\Delta_- = \min\operatorname{mod}(\Delta_-, b\Delta_+),$$

$1 \leqslant b \leqslant 3$ 为人为选用参数

$(\Delta_+)_j = u_{j+1} - u_j, (\Delta_-)_j = u_j - u_{j-1}$

$k = 1$ 中心格式，$k = 0$ Fromm 格式，$k = 1$ 迎风格式 $k = 1/3$ 三阶精度.

这个格式有较高的空间精度，称作 Monotonic Upstream Scheme for Consevation Laws 格式，简称 MUSCL 格式.

在以上格式中对流项采用迎风型，所以也叫迎风型 TVD 格式.Davis 导出了一种二阶 TVD 格式，它的数值通量由两部分组成，一项是通常的 L-W 二阶格式的数值通量，另一项数值耗散项，具有中心对称形式.后 Roe 将 Davis 的格式表达成容易分析的形式，导出了一些 Davis 没有发现的形式.后 Yee 对对称型 TVD 格式进行了深入研究和大量实际计算，证实了其效用，这种格式叫做 Yee-Roe-Davis 对称型二阶格式，它的具体形式如下：

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}), \tag{3-124}$$

其中 $\hat{f}_{j+\frac{1}{2}} = \frac{1}{2}[f(u_j) + f(u_{j+1}) + \psi_{j+\frac{1}{2}}]$,

$$\psi_{j+\frac{1}{2}} = -\frac{\Delta x}{\Delta t}\left[\left(\frac{\Delta t}{\Delta x_{aj+\frac{1}{2}}}\right)^2 q_{j+\frac{1}{2}} + Q(\lambda_{aj+\frac{1}{2}})(l_{j+1} - u_j - q_{j+\frac{1}{2}})\right],$$

$$q_{j+\frac{1}{2}} = \min\operatorname{mod}(u_j^n - u_{j-1}^n, u_{j+1} - u_j, u_{j+2}^n - u_{j+1}^n),$$

$$\bar{a}_{j+\frac{1}{2}} = \begin{cases} \dfrac{f(u_{j+1}) - f(u_j)}{u_{j+1} - u_j}, & u_{j+1} \neq u_j, \\ a(u_j), a = \dfrac{\partial f}{\partial u}, & u_{j+1} = u_j, \end{cases}$$

$$Q(\eta) = \begin{cases} |\eta|, & |\eta| > \varepsilon, \\ \dfrac{\eta^2 + \varepsilon^2}{2\varepsilon}, & |\eta| \leqslant \varepsilon. \end{cases}$$

这里可以看出,通量$\hat{f}$是由 L-W 通量加上一附加数值耗散项得到的,已经证明这一格式是 TVD 的,一般是二阶精度的.

需要指出 NND 格式也是 TVD 格式,其构造思想在前面章节中已经加以讨论了,不再细述.此外,前面利用限制器的方法中,Osher-Chakravarthy 格式最初是由二阶迎风格式改进得到的.还需说明的是以上所有格式中时间导数都用前向一阶精度格式.许多种格式中保持时间一阶导数项,从而格式为

$$\frac{\partial u}{\partial t}+\hat{f}_{j+\frac{1}{2}}-\hat{f}_{j-\frac{1}{2}}=0 \tag{3-125}$$

这叫半离散格式.各种格式都可以用这种半离散格式,利用它可在时间方向采用龙格-库塔法格式,从而提高时间方向的精度.最后,还可以进一步设计三阶或更高阶的 TVD 格式,这里不作详细介绍,可参阅有关文献.关于以上格式是否满足熵条件,均因过于复杂而省略.一般应参考原始的文献,一个较实用的方法是通过标准算例来验证.

3.5 ENO 格式

1985 年 Harten 和 Osher 等人进一步提出本质无振荡的 ENO 格式(Essentially non-oscillatary Scheme).它与 TVD 格式的区别在于

(1) TVD 格式减少总变差,强调消除振荡,而 ENO 强调消灭 $O(1)$ 阶的 Gibbs 振荡,但容许有 $O(h^r)$ 阶的附加虚拟振荡,这可以保留本来应当有的振荡,如一波通过激波时,这一波不因计算而消失.

(2) TVD 格式强调格式的单调性,而 ENO 更强调格式的自适应性.

下面简单讨论 ENO 格式构造的基本思想和格式的表示.

设求解的方程是

$$\frac{\partial U}{\partial t}+\frac{\partial F(U)}{\partial x}=0, \tag{3-126}$$

它的解可以表示为

$$U(x,t)=\mathrm{E}(t)U(x,0)=\mathrm{E}(t)U_0(x), \tag{3-127}$$

其中 $\mathrm{E}(t)$ 为一时间算子,这只是一个形式上的记号.引入平均算子

$$\overline{W}(x)=\frac{1}{h}\int_{x-\frac{h}{2}}^{x+\frac{h}{2}}w(y)\mathrm{d}y=(A_h w)(x), \tag{3-128}$$

其中 A_h 叫平均算子.于是对(3-126)式作用平均算子,有

$$\frac{\partial \overline{U}}{\partial t}+\frac{1}{h}\left[F\left(U\left(x+\frac{h}{2},t\right)\right)-F\left(U\left(x-\frac{h}{2},t\right)\right)\right],$$

对时间积分得

$$\overline{U}(x,t+\tau)=\overline{U}(x,t)-\frac{\tau}{h}\left[\hat{F}\left(x+\frac{h}{2},t;U\right)-\hat{F}\left(x-\frac{h}{2},t;U\right)\right]$$

其中
$$\hat{F}(x,t;U)=\frac{1}{\tau}\int_0^{\tau}F(U(x,t+\eta))\mathrm{d}\eta.$$

设 $x_j = jh, t_n = n\Delta t = n\tau$，故有

$$U_j^{n+1} = U_j^n - \frac{\tau}{h}[\hat{F}(x_{j+\frac{1}{2}}, t_n; U) - \hat{F}(x_{j-\frac{1}{2}}, t_n; U)], \tag{3-129}$$

其中

$$\bar{U}_j^n = \bar{U}(x_j, t_n) = \frac{1}{h}\int_{x_j-\frac{h}{2}}^{x_j+\frac{h}{2}} U(x, t_n)\mathrm{d}x, \tag{3-130}$$

称作网格区间平均. 以上方法是精确的，但不能直接使用，需用差分进行近似. 比如可写作

$$V^{n+1} = V_j^n - \frac{\tau}{h}(\bar{F}_{j+\frac{1}{2}} - \bar{F}_{j-\frac{1}{2}}) \equiv (\bar{E}_h(\tau) \cdot V^n) \tag{3-131}$$

其中 $\bar{E}_h(\tau)$ 为差分算子，$\bar{F}_{j+\frac{1}{2}}$ 也不同于(3-129)，而为

$$\hat{F}_{j+\frac{1}{2}} = \hat{F}(V_{j-k+1}, V_{j-k+2}, \cdots, V_{j+k}). \tag{3-132}$$

且

$$\hat{F}(V_j, V_j, \cdots, V_j) = \bar{F}(\bar{V}). \tag{3-133}$$

当 $U = V$ 时，$\hat{F}_{j+\frac{1}{2}}$ 和 $\hat{F}(x_{j+\frac{1}{2}}, t_n; U)$ 之间有 $O(h^r)$ 的差别，即

$$\hat{F}_{j+\frac{1}{2}} = \hat{F}(x_{j+\frac{1}{2}}, t_n; U) + d(x_{j+\frac{1}{2}})h^r + O(h^{r+1}),$$

其中 $d(x)$ 是一个利普希茨(Lipschitz)连续函数. 将以上各式组合，不难得到

$$\bar{U}(x_j, t_n + \tau) - \bar{E}_h(\tau)U(x_j, t_n) = \frac{\tau}{h}[d(x_{j+\frac{1}{2}}) - d(x_{j-\frac{1}{2}})]h^r + O(h^r).$$

可见差分解和 $U(x_j, t_n + \tau)$ 之间差值是 h^r 量级. 下面需要找出 $\hat{F}_{j+\frac{1}{2}}$(见(3-132)式)的具体形式，并要求不产生大于 h^r 的假振荡. Harten 的作法是：

(1) 用 V_j^n 作为 $\bar{U}^n(x_j)$ 的初值，设法再构造

$$V_h(x, t_n + 0) = R(x; V^n).$$

(2) 在小时间间隔内直接求解

$$V_h(\cdot, t) = E(t - t_n) \cdot V_h(\cdot, t_n + 0).$$

(3) 在网格内平均，得

$$V_j^{n+1} = \bar{V}_h(x_j; t_{n+1} - 0) = \frac{1}{h}\int_{x_j-\frac{h}{2}}^{x_j+\frac{h}{2}} V_h(x, t_{n+1} - 0)\mathrm{d}t,$$

这个 V_j^{n+1} 就是 $\bar{U}^{(n+1)}(x_j)$ 的近似值.

其中对 $R(x; V^n)$ 的要求为

$$R(x; \bar{w}) = w(x) + e(x)h^r + O(h^{r+1}),$$

$e(x)$ 为利普希茨连续函数，$\bar{R}(x, \bar{w}) = \bar{w}$ 以及 $TV(R(\cdot\ \bar{w})) \leqslant TV(w) + O(h^r)$.

以下介绍的方法是 Harten 的两种方法.

1) 原函数重构技术

设函数 $w(x)$ 的原函数 $W(x)$ 为

$$W(x) = \int_{x_0}^{x} w(x)\mathrm{d}x,$$

故
$$W(x_{j+\frac{1}{2}}) = \sum_{l=p}^{j} \overline{w}_l h_l,\quad \overline{w}_l = \overline{w}(x_l),\quad h_l = x_{l+1} - x_l.$$
利用 $W(x_{j+\frac{1}{2}})$ 构造 $W(x)$ 的分段 r 阶多项式 $H_r(x,w)$,它要满足

1° $H_r(x_{j+\frac{1}{2}}, W) = W(x_{j+\frac{1}{2}})$;

2° $\dfrac{d^k}{dx^k}H_r(x,W) = \dfrac{dk}{dx^k}W(x) + O(h^{r-k+1}), k = 0,1,2,\cdots,r$;

3° $TV(H_r(x,W)) \leqslant TV(W) + O(h^r)$.

有了 H_r,就可以取
$$R(x,\overline{w}) = \frac{d}{dx}H_r(x,W).$$
由于
$$\overline{R}(x,\overline{w}) = \frac{1}{\Delta x_j}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} R(x,\overline{w})dx = \frac{1}{\Delta x_j}[H_r(x_{j+\frac{1}{2}},W) - H_r(x_{j-\frac{1}{2}},W)]$$
$$= \frac{1}{\Delta x_j}[W(x_{j+\frac{1}{2}}) - W(x_{j-\frac{1}{2}})] = \overline{w}_j,$$
所以满足守恒性,又
$$\frac{d^l}{dx^l}R(x,\overline{w}) = \frac{d^{l+1}}{dx^{l+1}}H_r(x,W)$$
$$= \frac{d^{l+1}}{dx^{l+1}}W(x) + O(h^{r-l}) = \frac{d^l}{dx^l}w(x) + O(h^{r-l})$$
这说明精度要求满足.

2) 展开法

这里将 $w(x)$ 在 $x = x_0$ 处展开或改写为 $w(x+y)$ 后在 x 处展开,即得
$$w(x+y) = \sum_{k=0}^{\infty}\frac{1}{k!}y^k w^{(k)}(x),$$
于是
$$\overline{w}(x) = \sum_{k=0}^{\infty} a_k \Delta x^k w^{(k)}(x),$$
$$a_k = \frac{1}{\Delta x^{k+1}}\frac{1}{k!}\int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} y^k dy = \begin{cases}\dfrac{1}{2^k(k+1)!}, & k\ 为偶然,\\ 0, & k\ 为奇数.\end{cases}$$
对 $\overline{w}(x)$ 关于 x 求 l 次($l = 0,1,\cdots,r-1$)导数后得
$$\Delta x^l \overline{w}^{(l)}(x) = \sum_{k=0}^{\infty} a_k \Delta x^{k+l} w^{(k+l)}(x),$$
截断到 $O(\Delta x^r)$ 得
$$\Delta x^l \overline{w}^{(l)}(x) = \sum_{k=0}^{r-1-l} a_k \Delta x^{k+l}(x) + O(\Delta x^r).$$
这就得到 $\overline{w}(x)$ 及其各阶导数和 $w(x)$ 与各阶导数间的一个线性代数方程组

$$\overline{D}(x) = CD(x) + O(\Delta x^r), \tag{3-134}$$

其中

$$\overline{D}(x) = \begin{bmatrix} \overline{w}(x) \\ \Delta x\,\overline{w}^{(1)}(x) \\ \vdots \\ \Delta x^{r-l}\overline{w}^{(r-l)}(x) \end{bmatrix}, \qquad D(x) = \begin{bmatrix} w(x) \\ \Delta x w^{(1)}(x) \\ \vdots \\ \Delta x^{r-l} w^{(r-l)}(x) \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & a_1 & a_2 & \cdots & a_{r-1} \\ & 1 & a_1 & \cdots & a_{r-2} \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}.$$

有了给定的 $\overline{w} = \{\overline{w}_j\} = \{\overline{w}(x_j)\}$，再构造分段插值多项式 $H_m(x, \overline{w})$，满足以下条件：

1° $H_m(x_j, \overline{w}) = \overline{w}(x_j)$；

2° 在 $\overline{w}(x)$ 光滑点处，

$$\frac{d^k}{dx^k}H_m(x, \overline{w}) = \frac{d^k}{dx^k}\overline{w}(x) + O(\Delta x^{m+1-k}) \quad (k = 0, 1, \cdots, m);$$

3° $TV(H_m(x, \overline{w})) \leqslant TV(\overline{w}) + O(\Delta x^{m+1})$.

有了 $\overline{w}(x_j)$ 后可以将 $\overline{D}(x)$ 中各个元素计算出来：

$$\overline{D}_{0,j} = \overline{w}(x_j),$$

$$\overline{D}_{l,j} = \Delta x^l \min\bmod\left(\frac{d^l}{dx^l}H_m(x_j + 0, \overline{w}), \frac{d^l}{dx^l}H_m(x_j - 0, \overline{w})\right), l = 1, 2, \cdots, r-1,$$

其中
$$\frac{d^l}{dx^l}H_m(x_j \pm 0, \overline{w}) = \frac{d^l}{dx^l}\overline{w}(x_j) + O(\Delta x^{m+1-l}).$$

因此
$$\overline{D}_{l,j} = \Delta x^l \overline{w}^{(l)}(x_j) + O(\Delta x^r),$$

即
$$\overline{D}_j = \overline{D}(x_j) + O(\Delta x^r).$$

C 中各元素也是已知的，所以由(3-144)式略去 $O(\Delta x^r)$ 项后可解得 D_j，并由

$$D_{k,j} = \Delta x^k w^{(k)}(x_j) + O(\Delta x^r), k = 0, 1, \cdots, r-1$$

确定 $w^{(k)}(x_j)$. 于是

$$R(x, \overline{w}) = \sum_{k=0}^{r-1}\frac{1}{k!}D_{k,j}\left[\frac{x - x_j}{\Delta x}\right]^k, \quad |x - x_j| < \frac{\Delta x}{2}.$$

下面的问题是 H_r 及 H_m 的构造. 关于构造的细节可以参考有关的文献. 下面分别将两种不同方法构造的 H_r 及 H_m 代入上面的格式，最后得到的表达式. 由于通量计算是在空间进行的，所以可以用时间方向暂不离散的半离散格式.

用原函数构造 ENO，得 ENO-LF 格式.

将 $f(u)$ 分解为

$$f(u) = f^+(u) + f^-(u),$$

$$f^{+}(u)=\frac{1}{2}(f(u)+au),f^{-}(u)=\frac{1}{2}(f(u)-au),$$

其中 $a=\max\left|\frac{\mathrm{d}f}{\mathrm{d}u}\right|$，构造

$$P_{j+1/2}^{\pm}(x)=f^{\pm}(x)+O(\Delta x^{2m+1}),|x-x_{j+1/2}|\leqslant\frac{\Delta x}{2}.$$

$P_{j+1/2}^{\pm}(x)$ 算法如下：

(1) 取初值 $K_{\min}^{(0)}=K_{\max}^{(0)}=j,Q_{+}^{(0)}=f^{+}(x_j)$.

(2) 利用递推法由 $K_{\min}^{(n-1)}$、$K_{\max}^{(n-1)}$ 和 $Q_{+}^{(n-1)}(x)$ 计算 $K_{\min}^{(n)}$、$K_{\max}^{(n)}$ 和 $Q_{+}^{(n)}(x)$. 由于 $K_{\min}^{(n-1)}$、$K_{\max}^{(n-1)}$ 已知，则一个含有 n 个节点的插值区间 $[x_{K_{\min}^{(n-1)}},x_{K_{\max}^{(n-1)}}]$ 及在此模板上的 $(n-1)$ 阶插值函数 $Q_{+}^{(n-1)}(x)$ 已知.

为了扩大这个插值区间，有两个可能的方向.

向左　$[x_{K_{\min}^{(n-1)}-1},x_{K_{\max}^{(n-1)}}]$　(左一区间)，

向右　$[x_{K_{\min}^{(n-1)}},x_{K_{\max}^{(n-1)}+1}]$　(右一区间).

为确定向左、向右，要分别计算在左一区间和右一区间上的 $f^{+}(x)$ 的 n 阶差商. n 阶差商定义如下：

0 级差商　$\bar{f}^{+}(x_j)=f^{+}(x_j)$，

1 级差商　$\bar{f}^{+}(x_j,x_{j+1})=\dfrac{\bar{f}(x_{j+1})-\bar{f}^{+}(x_j)}{x_{j+1}-x_j}$，

2 级差商　$\bar{f}^{+}(x_j,x_{j+1},x_{j+2})=\dfrac{\bar{f}^{+}(x_{j+1}x_{j+2})-\bar{f}^{+}(x_jx_{j+1})}{x_{j+2}-x_j}$，

⋮

k 级差商为　$\bar{f}^{+}(x_j,x_{j+1},\cdots,x_{j+k})=\dfrac{\bar{f}^{+}(x_{j+1},x_{j+2}\cdots x_{j+k})-\bar{f}^{+}(x_j,x_{j+1}\cdots x_{j+k-1})}{x_{j+k}-x_j}$.

设

$$\begin{aligned}a^{(n)}&=f^{+}(x_{K_{\min}^{(n-1)}},\cdots,x_{K_{\max}^{(n-1)}+1})\quad\text{左一区间 }n\text{ 级差商}\\b^{(n)}&=f^{+}(x_{K_{\min}^{(n-1)}-1},\cdots,x_{K_{\max}^{(n-1)}})\quad\text{右一区间 }n\text{ 级差商}\end{aligned}\tag{3-135}$$

⋮

当 $|a^n|\geqslant|b^n|$ 时，左一区间为扩大后的区间，即

$$\begin{aligned}&K_{\min}^{(n)}=K_{\min}^{(n)}-1,\quad K_{\max}^{(n)}=K_{\max}^{(n-1)},\\&Q_{+}^{(n)}(x)=Q_{+}^{(n-1)}(x)+b^{(n)}\prod_{K_{\min}^{(n-1)}}^{K_{\max}^{(n-1)}}(x-x_k).\end{aligned}\tag{3-136}$$

当 $|a^n|<|b^n|$ 时，右一区间为扩大后的区间，即

$$\begin{aligned}&K_{\min}^{(n)}=K_{\min}^{(n-1)},\quad K_{\max}^{(n)}=K_{\max}^{(n-1)}+1,\\&Q_{+}^{(n)}(x)=Q_{+}^{(n-1)}(x)+a^{(n)}\prod_{K_{\min}^{(n-1)}}^{K_{\max}^{(n-1)}}(x-x_k).\end{aligned}\tag{3-137}$$

(3) $P^{+}_{j+1/2}(x) = Q^{(2m)}_{+}(x)$ 即为 $2m$ 阶插值多项式.

类似地构造 $P^{-}_{j+1/2}(x)$. 差别在于初值 $K^{(0)}_{\min} = K^{(0)}_{\max} = j+1$, 以后 f^{+} 均用 f^{-} 代替, 所有步骤同上. 于是

$$\hat{f}^{+}_{j+1/2} = P^{+}_{j+1/2}(x_{j+1/2}), \quad \hat{f}^{-}_{j+1/2} = P^{-}_{j+1/2}(x_{j+1/2}), \tag{3-138}$$

代入半离散的格式, 再用前面关于时间的 R-K 型格式, 即可解得 u^{n+1} 的值.

对于 $m = 0$,

$$P^{+}_{j+1/2}(x) = Q^{(0)}_{+}(x), \quad P^{-}_{j+1/2}(x) = Q^{(0)}_{-}(x),$$

$$\hat{f}^{+}_{j+1/2} = P^{+}_{j+1/2}(x_{j+1/2}) = f^{+}_{j} = \frac{1}{2}[f(u_j) + au_j],$$

$$\hat{f}^{-}_{j+1/2} = P^{-}_{j+1/2}(x_{j+1/2}) = f^{-}_{j+1} = \frac{1}{2}[f(u_{j+1}) - au_{j+1}],$$

$$\hat{f}_{j+1/2} = \hat{f}^{+}_{j+1/2} + \hat{f}^{-}_{j+1/2} = \frac{1}{2}[f(u_j) + f(u_{j+1}) - a(u_{j+1} - u_j)].$$

对于 $m = 1$, 引入记号　$mM(a,b) = \begin{cases} a, & |a| < |b|, \\ b, & |a| \geqslant |b|, \end{cases}$

$$Q^{(1)}_{+}(x) = f^{+}_{j} + (x - x_j)\,mM\left[\frac{\Delta_{j+1/2}f^{+}}{\Delta x}, \frac{\Delta_{j-1/2}f^{+}}{\Delta x}\right],$$

$$Q^{(1)}_{-}(x) = f^{-}_{j+1} + (x - x_{j+1})\,mM\left[\frac{\Delta_{j+3/2}f^{-}}{\Delta x}, \frac{\Delta_{j+1/2}f^{-}}{\Delta x}\right],$$

得

$$\begin{cases} \hat{f}^{+}_{j+1/2} = P^{+}_{j+1/2}(x_{j+1/2}) = f^{+}_{j} + \dfrac{1}{2}mM(\Delta_{j+1/2}f^{+}, \Delta_{j-1/2}f^{+}), \\ \hat{f}^{-}_{j+1/2} = P^{-}_{j+1/2}(x_{j+1/2}) = f^{-}_{j+1} - \dfrac{1}{2}mM(\Delta_{j+3/2}f^{-}, \Delta_{j+1/2}f^{-}). \end{cases}$$

对于 $m = 2$,

$$Q^{(2)}_{+}(x) = Q^{(1)}(x) + \begin{cases} \dfrac{(x - x_j)(x - x_{j+1})}{2\Delta x^2}mM(\Delta_{j+3/2}f^{+} - \Delta_{j+1/2}f^{+}, \Delta_{j+1/2}f^{+} - \Delta_{j+1/2}f^{+}) \\ \qquad\qquad (\text{当 } |\Delta_{j+1/2}f^{+}| < |\Delta_{j-1/2}f^{+}|), \\ \dfrac{(x - x_j)(x - x_{j-1})}{2\Delta x^2}mM(\Delta_{j+1/2}f^{+} - \Delta_{j-1/2}f^{+}, \Delta_{j-1/2}f^{+} - \Delta_{j-3/2}f^{+}) \\ \qquad\qquad (\text{当 } |\Delta_{j+1/2}f^{+}| \geqslant |\Delta_{j-1/2}f^{+}|), \end{cases}$$

$$Q^{(2)}_{-}(x) = Q^{(1)}(x) + \begin{cases} \dfrac{(x - x_{j+1})(x - x_{j+2})}{2\Delta x^2}mM(\Delta_{j+5/2}f^{-} - \Delta_{j+3/2}f^{-}, \Delta_{j+3/2}f^{-} - \Delta_{j+1/2}f^{-}) \\ \qquad\qquad (\text{当 } |\Delta_{j+3/2}f^{-}| < |\Delta_{j+1/2}f^{-}|), \\ \dfrac{(x - x_{j+1})(x - x_j)}{2\Delta x^2}mM(\Delta_{j+3/2}f^{-} - \Delta_{j+1/2}f^{-}, \Delta_{j+1/2}f^{-} - \Delta_{j-1/2}f^{-}) \\ \qquad\qquad (\text{当 } |\Delta_{j+3/2}f^{-}| \geqslant |\Delta_{j+1/2}f^{-}|). \end{cases}$$

对于 $m=3$,

$$\hat{f}^{+}_{j+1/2}=P^{+}_{j+1/2}(x_{j+1/2})-\frac{\Delta x^2}{24}\frac{\partial^2}{\partial x^2}P^{+}_{j+1/2}(x_{j+1/2})$$

$$=f^{+}_{j}+\frac{1}{2}mM(\Delta_{j+1/2}f^{+},\Delta_{j-1/2}f^{+})+$$

$$\begin{cases}(-\frac{1}{8}-\frac{1}{24})mM(\Delta_{j+3/2}f^{+}-\Delta_{j+1/2}f^{+},\Delta_{j+1/2}f^{+}-\Delta_{j-1/2}f^{+}) \\ \qquad (\text{当 } |\Delta_{j+1/2}f^{+}|<|\Delta_{j-1/2}f^{+}|), \\ (\frac{3}{8}-\frac{1}{24})mM(\Delta_{j+1/2}f^{+}-\Delta_{j-1/2}f^{+},\Delta_{j-1/2}f^{+}-\Delta_{j-3/2}f^{+}) \\ \qquad (\text{当 } |\Delta_{j+1/2}f^{+}|\geqslant|\Delta_{j-1/2}f^{+}|),\end{cases}$$

$$\hat{f}^{-}_{j+1/2}=P^{-}_{j+1/2}(x_{j+1/2})-\frac{\Delta x^2}{24}\frac{\partial^2}{\partial x^2}P^{-}_{j+1/2}(x_{j+1/2})$$

$$=f^{-}_{j}-\frac{1}{2}mM(\Delta_{j+3/2}f^{-},\Delta_{j+1/2}f^{-})+$$

$$\begin{cases}(\frac{3}{8}-\frac{1}{24})mM(\Delta_{j+5/2}f^{-}-\Delta_{j+3/2}f^{-},\Delta_{j+3/2}f^{-}-\Delta_{j+1/2}f^{-}) \\ \qquad (\text{当 } |\Delta_{j+3/2}f^{-}|<|\Delta_{j+1/2}f^{-}|), \\ (-\frac{1}{8}-\frac{1}{24})mM(\Delta_{j+3/2}f^{-}-\Delta_{j+1/2}f^{-},\Delta_{j+1/2}f^{-}-\Delta_{j-1/2}f^{-}) \\ \qquad (\text{当 } |\Delta_{j+3/2}f^{-}|\geqslant|\Delta_{j+1/2}f^{-}|).\end{cases}$$

除以上 ENO-LF 方法外，还存在一些其它的 ENO 方法.

3.6　气体动力学格式

在前面讨论的格式中，都是从欧拉方程出发，并受黎曼解的启发而得到的. 在这里必须经常关注所得的弱解是否满足熵条件，而熵又不一定是物理意义上的熵，而是由数学家杜撰出来的一个函数. 如前所看到的，关于熵函数和熵通量组成的熵对，尚有不少疑虑，熵条件的检验也是十分困难的事. 本节要介绍的由气体动力学发展得到的格式，则有全然不同的构思，这里从微观方程出发，通过取矩而得到微观量与宏观量的关系，进而建立宏观量的差分量之间的关系. 该方法由于是从玻耳兹曼(Boltzmann)方程出发的，它总满足熵条件，又由于它有较深的物理内涵，所以可以计算各种物理问题，而且计算更稳健(robust).

用微观方法描述流体的流动，首先要引入分布函数的概念. 流体分子的运动是用其不同时刻所在的空间位置及其速度来表示，设每个分子的质量都是 m，在 t 时刻位于 $\boldsymbol{r}$ 处的分子速度为 $\boldsymbol{u}$，单位体积内的分子数为 n，则定义

$$mn=f(\boldsymbol{r},t,\boldsymbol{u}) \tag{3-139}$$

为分布函数，不难看出 f 实际有概率的含义. 对相空间(这里就是速度空间)积分，可得

$$\rho = \int f \mathrm{d}u \mathrm{d}v \mathrm{d}w, \tag{3-140}$$

积分域为整个相空间，积分值 $\rho = (\boldsymbol{r}, t)$. 这就将分布函数和宏观密度量联系起来了. 上面讨论的事实上是单原子的情况，所以确定其运动状态只有三个速度分量，对于一般的分子，还有若干内自由度（如方位，相互间的距离等），记它们作 $\xi_1, \xi_2, \cdots, \xi_N$，这时分布函数 f 应为

$$mn = f(\boldsymbol{r}, t, \boldsymbol{u}, \xi_1, \cdots, \xi_N). \tag{3-141}$$

在整个相空间内积分得

$$\rho(\boldsymbol{r}, t) = \int f(\boldsymbol{r}, t, \boldsymbol{u}, \xi_1, \cdots, \xi_N) \mathrm{d}u \mathrm{d}v \mathrm{d}w \mathrm{d}\xi_1 \cdots \mathrm{d}\xi_N \tag{3-142}$$

$$= \int f(\boldsymbol{r}, t, \boldsymbol{u}, \xi) \mathrm{d}\Xi,$$

其中 $\mathrm{d}\Xi = \mathrm{d}u \mathrm{d}v \mathrm{d}w \mathrm{d}\xi_1 \cdots \mathrm{d}\xi_N$，　$\xi = (\xi_1, \xi_2, \cdots, \xi_N)$，$N$ 为自由度数.

根据统计力学理论知道，每个自由度的统计平均能量都应当是 $\frac{1}{2}kT$，其中 k 为玻耳兹曼常数，T 为绝对温度. 由此知道

$$C_V = \frac{\frac{1}{2}kT(N+3)}{mT} = \frac{N+3}{2}R, \quad R = \frac{k}{m} \text{（气体常数）}. \tag{3-143}$$

而 $C_p = C_V + R$，故比热 γ 为

$$\gamma = \frac{C_p}{C_V} = \frac{\frac{N+3}{2}+1}{\frac{N+3}{2}} = \frac{N+5}{N+3}. \tag{3-144}$$

在以后讨论的问题中，都近似的认为气体分子运动离平衡态分布不是很远，而平衡态的分布用麦克斯韦尔 - 玻耳兹曼（Maxwell-Boltzmann）分布来表示，这是

$$g = \rho\left(\frac{\lambda}{\pi}\right)^{\frac{N+3}{2}} \mathrm{e}^{-\lambda[(u-U)^2+(v-V)^2+(w-W)^2+\xi_1^2+\cdots+\xi_K^2]} \tag{3-145}$$

其中 ρ 为流体密度，$\lambda = \frac{m}{2kT}$ 是温度的函数，U、V、W 是流体的宏观流动速度. ρ、$\lambda(T)$、U、V、W 都是 $(\boldsymbol{r}, t)$ 的函数，不难看出：

$$\begin{bmatrix} \rho \\ \rho U \\ \rho V \\ \rho W \\ \rho\varepsilon \end{bmatrix} = \int_{-\infty}^{+\infty} g \begin{bmatrix} 1 \\ u \\ v \\ w \\ \frac{1}{2}(u^2+v^2+w^2+\xi_1^2+\cdots+\xi_N^2) \end{bmatrix} \mathrm{d}u \mathrm{d}v \mathrm{d}w \mathrm{d}\xi_1 \cdots \mathrm{d}\xi_N,$$

其中

$$\rho\varepsilon = \frac{1}{2}\rho\left[U^2 + V^2 + W^2 + \frac{K+3}{2\lambda}\right],$$

即为单位流体体积的动能和势能. 详细的关于平衡态分布函数的取矩积分公式可

以在本节末的附录中找到.此处还可以由此求得相应的内自由度$\frac{-3\gamma+5}{\gamma-1}=N$

一维和二维问题有些不同,可见表3-1.

表3-1

	一维	二维	三维	d 维
K	$N+2$	$N+1$	N	$N+3-d$
与 γ 关系	$\frac{3-\gamma}{\gamma-1}$	$\frac{4-2\gamma}{\gamma-1}$	$\frac{5-3\gamma}{\gamma-1}$	$\frac{2+d-d\gamma}{\gamma-1}$
$\rho\varepsilon$	$\frac{\rho}{2}\left(U^2+\frac{K+1}{2\lambda}\right)$	$\frac{\rho}{2}\left(U^2+V^2+\frac{K+2}{2\lambda}\right)$	$\frac{\rho}{2}\left(U^2+V^2+W^2+\frac{K+3}{2\lambda}\right)$	$\frac{\rho}{2}\left(\sum_{i=1}^{d}U_i^2+\frac{K+d}{2\lambda}\right)$

此外由统计力学知

$$\lambda=\frac{m}{2kT},\quad p=nkT=\frac{\rho}{m}k\frac{m}{2k\lambda}=\frac{\rho}{2\lambda}$$

这里注意到平衡态的分布函数与当地的宏观量有一一对应的关系.但是一般情况,流体并不总是处于平衡状态,比如通过激波时就不处于平衡态.一般非平衡态下的分布函数记为f,它的演变过程满足玻耳兹曼方程,即

$$f_t+u_i f_{x_i}+a_i f_{u_i}=Q(f,f). \tag{3-146}$$

其中u_i、a_i分别是分子运动的速度及作用于质点上的体积力在i方向上的分量,$Q(f,f)$是质点碰撞项.由于在碰撞过程中,质点的质量不发生变化,而且动量和能量也都是不变的,所以应当有

$$\int\Psi_\alpha Q(f,f)\mathrm{d}\Xi=0, \tag{3-147}$$

其中$\mathrm{d}\Xi=\mathrm{d}u\mathrm{d}v\mathrm{d}w\mathrm{d}\xi_1\cdots\mathrm{d}\xi_K$,$K$即前面讨论的$N+3-d$,而

$$\Psi_\alpha=\left(1,u,v,w,\frac{1}{2}(u^2+v^2+w^2+\xi_1^2+\cdots+\xi_K^2)\right)^{\mathrm{T}}.$$

可以证明由玻耳兹曼方程计算得的粘性应力张量、热通量等与纳维-斯托克斯方程是相同的,也就是说从玻耳兹曼方程出发得到的宏观量满足纳维-斯托克斯方程.1954年由Bhatnagar-Gross-Krook提出的BGK模型就是一种最常用的模式,它假定$Q\propto(f-g)$.由于碰撞的作用是使分布由非平衡态向平衡态过渡,因此使之变为平衡的速度应当与分布偏离平衡态的程度成正比.另一方面又设$Q\propto 1/\tau$,即碰撞次数越频繁,则向平衡态过渡的速度越快,其中τ是质点先后两次碰撞之间的时间间隔的统计平均值.根据以上假定,有

$$Q\propto\frac{(f-g)}{\tau}. \tag{3-148}$$

于是利用这个BGK模型得到的相应玻耳兹曼方程为

$$f+u_i f_{x_i}=-\frac{(f-g)}{\tau}. \tag{3-149}$$

将其中外力作用项$u_i f_{x_i}$略去不计,即不计作用于质点上的作用力,由(3-147)式可知,有

$$\int \frac{(f-g)}{\tau}\Psi_a \mathrm{d}u\mathrm{d}v\mathrm{d}w\mathrm{d}\xi = 0. \tag{3-150}$$

设 τ 是一个常数，则(3-149) 式的通解为

$$f(x_i,t,u_i,\xi) = \frac{1}{\tau}\int_{t_0}^{t} g(x_i - u_i(t-t'),t',u_i,\xi)\mathrm{e}^{-\frac{t-t'}{\tau}}\mathrm{d}t' + \mathrm{e}^{-\frac{t-t_0}{\tau}} f_0(x_i - u_i(t-t_0),t_0,u_i,\xi). \tag{3-151}$$

这个方程解的正确性可以将它代入(3-149) 式加以验证. f_0 是 $t = t_0$ 时的分布函数，g 是 (x,t) 时的平衡态.

另一个重要的性质是玻耳兹曼方程满足 H 定理，即熵增原理。这个证明可以在应纯同的"气体输运理论"一书中找到. 这里讨论的是当玻耳兹曼方程中的碰撞项用 BGK 模型时，H 定理仍然是满足的. 首先要说明一下什么是 H 定理. 由于分布函数实际上是一个概率分布函数，所以由统计力学中知道，系统的熵可以用

$$H = \int f \ln f \mathrm{d}\Xi, \quad S = -H \tag{3-152}$$

来计算，而在 i 方向的熵流则为

$$H_i = \int u_i f \ln f \mathrm{d}\Xi, \quad S_i = -H_i. \tag{3-153}$$

由熵增定理可知

$$\frac{\partial H}{\partial t} + \frac{\partial H_i}{\partial x_i} \leqslant 0,$$

所以由方程(3-149) 和(3-152)、(3-153) 式有

$$\frac{\partial H}{\partial t} + \frac{\partial H_i}{\partial x_i} = \int\left(\frac{\partial f}{\partial t} + u_i\frac{\partial f_i}{\partial x_i}\right)(1+\ln f)\mathrm{d}\Xi = \int \frac{g-f}{\tau}(1+\ln f)\mathrm{d}\Xi.$$

考虑到(3-150) 式中第一式(即 Ψ_a 中 1 的这一项)，有

$$\frac{\partial H}{\partial t} + \frac{\partial H_i}{\partial x_i} = \frac{1}{\tau}\int (g-f)\ln f\mathrm{d}\Xi = \frac{1}{\tau}\int (g-f)(\ln f - \ln g)\mathrm{d}\Xi + \frac{1}{\tau}\int (g-f)\ln g\mathrm{d}\Xi,$$

又考虑 g 的表达式和(3-150) 式，可知

$$\int (g-f)\ln g\mathrm{d}\Xi = 0,$$

因此

$$\frac{\partial H}{\partial t} + \frac{\partial H_i}{\partial x_i} = \frac{1}{\tau}\int (g-f)(\ln f - \ln g)\mathrm{d}\Xi \leqslant 0 \tag{3-154}$$

(当 $g \neq f$ 时由于 g、f 都是正的，所以 $g > f$ 时必有 $\ln g > \ln f$，反之亦然. 所以上述不等式必然成立). 由上分析得知，由 BGK 模式得到的结果必然满足熵增原理，所以在以后的分析讨论中完全不必考虑熵条件的问题，这一点是比前面讨论的格式优越的地方. 因为它的出发点是物理问题本身而不是从数学上专门构造一个熵函数，后者并不能完全反映物理问题本身，而且也存在着物理上难以完善的地方. 有

了以上 BGK 模式，下面来讨论由此建立的有限体积的格式. 记

$$W = (\rho, \rho U, \rho\varepsilon)^{\mathrm{T}} = \int \Psi_a f \mathrm{d}u \mathrm{d}\xi,$$

$$F(W) = (F_\rho, F_{\rho U}, F_{\rho\varepsilon})^{\mathrm{T}} = \int u \Psi_a f \mathrm{d}u \mathrm{d}\xi,$$

其中 W 为守恒量，$F(W)$ 是 W 的通量，$\Psi_a = (1, u, \frac{1}{2}(u^2 + \xi^2))^{\mathrm{T}}$. 由玻耳兹曼方程在 $(x_{j-1/2}, x_{j+1/2})(t^n, t^n + \Delta t)$ 上积分，得

$$\int (f_t + u f_x) \Psi_a \mathrm{d}u \mathrm{d}\xi \mathrm{d}x \mathrm{d}t = \int Q(f, f) \Psi_a \mathrm{d}u \mathrm{d}\xi \mathrm{d}x \mathrm{d}t.$$

右端项为零，所以得到

$$\int f_t \Psi_a \mathrm{d}u \mathrm{d}\xi \mathrm{d}x \mathrm{d}t + \int u f_x \Psi_a \mathrm{d}u \mathrm{d}\xi \mathrm{d}x \mathrm{d}t = 0$$

或

$$W_j^{n+1} - W_j^n + \frac{1}{\Delta x} \int_{t^n}^{t^n + \Delta t} (F_{j+1/2} - F_{j-1/2}) \mathrm{d}t = 0, \tag{3-155}$$

这就是 BGK 有限体积法格式. 对于多维即为

$$W_{ijk}^{n+1} - W_{ijk}^n + \frac{1}{\Delta x} \int_{t^n}^{t^n + \Delta t} (F_{x+1/2, jk} - F_{xi-1/2, jk}) \mathrm{d}t + \frac{1}{\Delta y} \int_{t^n}^{t^n + \Delta t} (F_{y_{ij}+k/2} - F_{y_{ij}-k/2}) \mathrm{d}t +$$

$$\frac{1}{\Delta z} \int_{t^n}^{t^n + \Delta t} (F_{xijk+1/2} - F_{xijk-1/2}) \mathrm{d}t = 0. \tag{3-156}$$

下面以一维为例说明网格连结处守恒量通量的计算. 在构造一阶精度的格式中，设在各网格$(x_{j-1/2}, x_{j+1/2})$内，分布函数是一个常数，而且是平衡态，即

$$f_0(\bar{x}, 0) = \begin{cases} g_l, & \bar{x} \leqslant 0, \\ g_r, & \bar{x} > 0, \end{cases} \quad \bar{x} = x - x_{j+1/2}, \tag{3-157}$$

其中 g_l、g_r 为平衡态，由 x_j 及 x_{j+1} 处的守恒量大小直接确定. 上式又可写为

$$f_0(\bar{x}, 0) = g_l[1 - H(\bar{x})] + g_r H(\bar{x}). \tag{3-158}$$

由于分布函数满足方程(3-149)，所以满足其通解形式(3-151). 设 $g_0(0,0)$ 为 $\bar{x} = 0$、$t = 0$ 时由于初始碰撞产生的平缓初始状态，这样在 $x = 0$ 处，由(3-151)式通解可以得到

$$f(\bar{x} = 0, t) = (1 - \mathrm{e}^{-t/\tau}) g_0 + \mathrm{e}^{-t/\tau} [g_l H(u) + g_r(1 - H(u))]. \tag{3-159}$$

不难看出，当 $\tau \to 0$ 时

$$f(\bar{x} = 0, t) = g_0,$$

而 $\tau \to \infty$ 时，

$$f(\bar{x} = 0, t) = f_0(\bar{x} = 0, t) = g_l H(u) + g_r(1 - H(u)).$$

为了确定 g_0，可以利用相容条件，并令 $\bar{x} = 0$，即

$$\int_{-\infty}^{+\infty} \Psi_a g_0 \mathrm{d}u \mathrm{d}\xi = \int \Psi_a f_0(-ut) \mathrm{d}u \mathrm{d}\xi$$

$$= \int_{u>0} \Psi_a g_l \mathrm{d}u\mathrm{d}\xi + \int_{u<0} \Psi_a g_r \mathrm{d}u\mathrm{d}\xi.$$

从图 3-8 可以看到，虚线构成 f_0，实线构成 g_0.

另外(3-159) 式可以写作

$$\begin{cases} f = (1-\eta)g_0 + \eta f_0, \\ \eta = \mathrm{e}^{-t/\tau}. \end{cases} \tag{3-160}$$

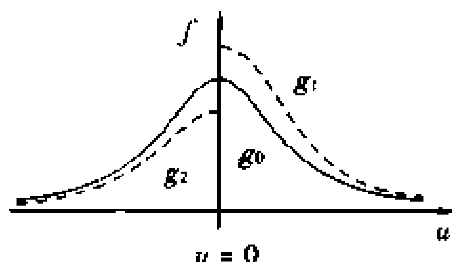


图 3-8

由以上表达式就可以求出一阶有碰撞玻耳兹曼格式. 其过程为

(1) 给出某一时刻的守恒量

$$\{\rho_j^n, \rho_j^n U_j^n, \rho_j^n \varepsilon_j^n\},$$

则得$(x_{j-1/2})x_{j+1/2}$,网络内的平衡态分布函数为

$$g_j = \rho_j \left(\frac{\lambda_j}{\pi}\right)^{\frac{k+1}{2}} \mathrm{e}^{-\lambda_j[(u-U_j)^2+\xi^2]},$$

其中

$$\lambda_j = \frac{k+1}{4} \frac{\rho_j}{\rho_j \varepsilon_j - \frac{1}{2}\rho_j U_j^2}.$$

(2) 由 f_0 计算 $x_{j+1/2}$ 处的数值通量

$$\begin{bmatrix} F^0_{\rho,j+1/2} \\ F^0_{\rho U,j+1/2} \\ F^0_{\rho\varepsilon,j+1/2} \end{bmatrix} = \int u\Psi_a f_0 \mathrm{d}u\mathrm{d}\xi = \int_{u>0} u\Psi_a g_l \mathrm{d}u\mathrm{d}\xi = \int_{u<0} u\Psi_a g_r \mathrm{d}u\mathrm{d}\xi$$

$$= \rho_j \begin{bmatrix} \frac{U_j}{2}\mathrm{erfc}(-\sqrt{\lambda_j}U_j) + \frac{1}{2}\frac{\mathrm{e}^{-\lambda_j U_j^2}}{2\sqrt{\pi\lambda_j}} \\ \left(\frac{U_j^2}{2} + \frac{1}{4\lambda_j}\right)\mathrm{erfc}(-\sqrt{\lambda_j}U_j) + \frac{U_j}{2}\frac{\mathrm{e}^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}} \\ \left(\frac{U_j^3}{4} + \frac{k+3}{8\lambda_j}U_j\right)\mathrm{erfc}(-\sqrt{\lambda_j}U_j) + \left(\frac{U_j^2}{4} + \frac{k+2}{8\lambda_j}\right)\frac{\mathrm{e}^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}} \end{bmatrix} +$$

$$\rho_{j+1} \begin{bmatrix} \frac{U_{j+1}^2}{2}\mathrm{erfc}(\sqrt{\lambda_{j+1}}U_{j+1}) - \frac{1}{2}\frac{\mathrm{e}^{-\lambda_{j+1}U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}} \\ \left(\frac{U_{j+1}^2}{2} + \frac{1}{4\lambda_{j+1}}\right)\mathrm{erfc}(\sqrt{\lambda_{j+1}}U_{j+1}) - \frac{U_{j+1}}{2}\frac{\mathrm{e}^{-\lambda_{j+1}U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}} \\ \left(\frac{U_{j+1}^3}{4} + \frac{k+3}{8\lambda_{j+1}}U_{j+1}\right)\mathrm{erfc}(\sqrt{\lambda_{j+1}}U_{j+1}) - \left(\frac{U_{j+1}^2}{4} + \frac{k+2}{8\lambda_{j+1}}\right)\frac{\mathrm{e}^{-\lambda_{j+1}U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}} \end{bmatrix}. \tag{3-161}$$

(3) 计算 $x_{j+1/2}$ 处的守恒量

$$\begin{bmatrix}\rho_{j+1/2}\\ \rho_{j+1/2}U_{j+1/2}\\ \rho_{j+1/2}\varepsilon_{j+1/2}\end{bmatrix} = \int \Psi_a g_0 \mathrm{d}u\mathrm{d}\xi = \int_{u>0}\Psi_a g_j \mathrm{d}u\mathrm{d}\xi + \int_{u<0}\Psi_a g_{j+1}\mathrm{d}u\mathrm{d}\xi$$

$$= \rho_j\begin{bmatrix}\frac{1}{2}\mathrm{erfc}(-\sqrt{\lambda_j}U_j)\\ \frac{1}{2}U_j\mathrm{erfc}(-\sqrt{\lambda_j}U_j)+\frac{1}{2}\frac{\mathrm{e}^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}}\\ \frac{1}{2}\left(\frac{U_j^2}{2}+\frac{k+1}{4\lambda_j}\right)\mathrm{erfc}(-\sqrt{\lambda_j}U_j)+\frac{U_j}{4}\frac{\mathrm{e}^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}}\end{bmatrix}+$$

$$\rho_{j+1}\begin{bmatrix}\frac{1}{2}\mathrm{erfc}(\sqrt{\lambda_{j+1}}U_{j+1})\\ \frac{1}{2}U_{j+1}\mathrm{erfc}(\sqrt{\lambda_{j+1}}U_{j+1})-\frac{1}{2}\frac{\mathrm{e}^{-\lambda_{j+1}U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}}\\ \frac{1}{2}\left(\frac{U_{j+1}^2}{2}+\frac{k+1}{4\lambda_{j+1}}\right)\mathrm{erfc}(\sqrt{\lambda_{j+1}}U_{j+1})-\frac{U_{j+1}}{4}\frac{\mathrm{e}^{-\lambda_{j+1}U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}}\end{bmatrix}. \tag{3-162}$$

(4) 由 g_0 计算 $x_{j+1/2}$ 处的守恒量的通量,

$$\begin{bmatrix}F^{g_0}_{\rho,j+1/2}\\ F^{g_0}_{\rho U,j+1/2}\\ F^{g_0}_{\rho\varepsilon,j+1/2}\end{bmatrix} = \int_{-\infty}^{+\infty} u\Psi_a g_0\mathrm{d}u\mathrm{d}\xi = \rho_{j+1/2}\begin{bmatrix}U_{j+1/2}\\ U_{j+1/2}^2+\frac{1}{2\lambda_{j+1/2}}\\ \frac{1}{2}U_{j+1/2}^3+\frac{k+3}{4\lambda_{j+1/2}}U_{j+1/2}\end{bmatrix}. \tag{3-163}$$

(5) 最后得通过 $x_{j+1/2}$ 处的通量

$$F_{w,j+1/2} = \begin{bmatrix}F_{\rho,j+1/2}\\ F_{\rho U,j+1/2}\\ F_{\rho\varepsilon,j+1/2}\end{bmatrix} = (1-\mathrm{e}^{-t/\tau})\begin{bmatrix}F^{g_0}_{\rho,j+1/2}\\ F^{g_0}_{\rho U,j+1/2}\\ F^{g_0}_{\rho\varepsilon,j+1/2}\end{bmatrix} + \mathrm{e}^{-t/\tau}\begin{bmatrix}F^{0}_{\rho,j+1/2}\\ F^{0}_{\rho U,j+1/2}\\ F^{0}_{\rho\varepsilon,j+1/2}\end{bmatrix} \tag{3-164}$$

类似地可以算得 $F_{w,j-1/2}$,代入(3-155)式可以算得 W^{n+1}. 这样得到的格式就是一阶格式. 关于格式的保正性,可以证明 $\eta = \mathrm{e}^{-t/\tau} = 1$ 时保正性是没有问题的. 但 $0 < \eta < 1$ 时的保正性,证明比较困难. 计算表明,当马赫数 $M < 15$ 时,即使 $\eta = 0$ 也具有保正性. 实际使用时,间断处一般选 $\eta \sim 1/2$,而且看来,很有可能对一般的 M 数都有保正性,即使 $M = 10^4$,但 η 需适当选用.

为了提高精度,下面进一步讨论二阶精度的情况. 这时和上一节一样,在 $(x_{j-1/2}, x_{j+1/2})$ 间隔内守恒量有变化为

$$W_j(x) = W_j + L(s_{j+}, s_{j-})(x - x_j),\ x_{j-1/2} \leqslant x \leqslant x_{j+1/2}, \tag{3-165}$$

其中 $L(s_{j+}, s_{j-})$—$L(r, s)$ 为一限制器,有以下几种.

(1) Van Leer 限制器

$$L(r,s)=S(r,s)\frac{2|r||s|}{|r|+|s|},\tag{3-166}$$

$$S(a,b)=\begin{cases}1, & a,b>0,\\ -1, & a,b<0,\\ 0, & a>0>b \quad or \quad b>0>a.\end{cases}$$

(2) MUSCL 限制器

$$L(r,s)=S(r,s).$$

$$\min\left(\frac{1}{2}|r+s|+s,2|r|,2|s|\right).$$

由上可知

$$W_{j+1/2}(x)=W(x_{j+1/2})=W_j+\frac{h}{2}L(s_{j+},s_{j-}),$$

其中
$$s_{j+}=\frac{F_{j+1}-F_j}{x_{j+1}-x_j},\quad s_{j-}=\frac{F_j-F_{j-1}}{x_j-x_{j-1}}.\tag{3-167}$$

F 为某一守恒量($\rho,\rho U$ 或 ρE).

有了以上计算,可以确定 g_r、g_l 所对应的守恒量是 $W^+_{j+1/2}$ 和 $W^-_{j+1/2}$.

计及守恒量在网格内的变化,则有

$$f_0=\begin{cases}g_l(1+a_l(x-x_{j+1/2})), & x\leqslant x_{j+1/2},\\ g_r(1+a_r(x-x_{j+1/2})), & x>x_{j+1/2},\end{cases}\tag{3-168}$$

和
$$g=g_0(1+(1-H(\bar{x}))\bar{a}_l(\bar{x})+H(\bar{x})\bar{a}_r(\bar{x})+\bar{A}t),\tag{3-169}$$

其中 $\bar{A}t$ 是由 g 在时间方向的变化得到,且有

$$\begin{aligned}
a_l&=a_{l1}+a_{l2}u+a_{l3}\frac{1}{2}(u^2+\xi^2)=a_{l\alpha}\Psi_\alpha,\\
a_r&=a_{r1}+a_{r2}u+a_{r3}\frac{1}{2}(u^2+\xi^2)=a_{r\alpha}\Psi_\alpha,\\
\bar{a}_l&=\bar{a}_{l1}+\bar{a}_{l2}u+\bar{a}_{l3}\frac{1}{2}(u^2+\xi^2)=\bar{a}_{l\alpha}\Psi_\alpha,\\
\bar{a}_r&=\bar{a}_{r1}+\bar{a}_{r2}u+\bar{a}_{r3}\frac{1}{2}(u^2+\xi^2)=\bar{a}_{r\alpha}\Psi_\alpha,\\
\bar{A}&=A_1+A_2u+A_3\frac{1}{2}(u^2+\xi^2)=\bar{A}_\alpha\Psi_\alpha,\\
\alpha&=1,2,3,\quad \Psi_1=1,\Psi_2=u,\Psi_3=\frac{1}{2}(u^2+\xi^2).
\end{aligned}\tag{3-170}$$

由(3-167)式可计算得

左侧　　　　　　　　　　右侧

$$\begin{bmatrix}\rho_l\\ \rho_lU_l\\ \rho\epsilon_l\end{bmatrix}=\begin{bmatrix}\rho_j(x_{j+1/2})\\ \rho_jU_j(x_{j+1/2})\\ \rho\epsilon_j(x_{j+1/2})\end{bmatrix},\quad \begin{bmatrix}\rho_{j+1}(x_{j+1/2})\\ \rho_{j+1}U_{j+1}(x_{j+1/2})\\ \rho_{j+1}\epsilon_{j+1}(x_{j+1/2})\end{bmatrix}=\begin{bmatrix}\rho_r\\ \rho_rU_r\\ \rho_r\epsilon_r\end{bmatrix}.$$

由此可以确定 g_l 和 g_r,参见图 3-9. 然后下面来确定 f_0 与守恒量的关系.

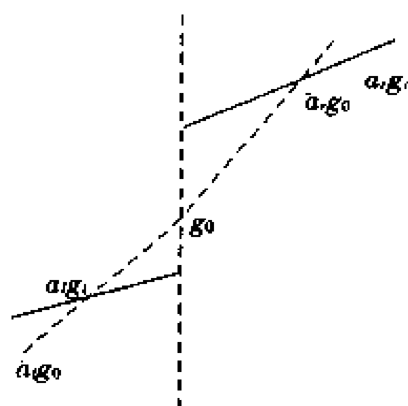


图 3-9

$$
\begin{cases}
\boldsymbol{W}_l = \int \boldsymbol{\Psi}_\alpha g_l \mathrm{d}\varepsilon = \begin{bmatrix} \rho_j(x_{j+1/2}) \\ \rho_j U_j(x_{j+1/2}) \\ \rho\varepsilon_j(x_{j+1/2}) \end{bmatrix}, \\
\boldsymbol{F}_{W_l} = \int \boldsymbol{\Psi}_\alpha g_r \mathrm{d}\varepsilon = \begin{bmatrix} \dfrac{\rho_j(x_{j+1/2})-\rho_j(x_j)}{x_{j+1/2}-x_j} \\ \dfrac{\rho_j V_j(x_{j+1/2})-\rho_j V_j(x_j)}{x_{j+1/2}-x_j} \\ \dfrac{\rho\varepsilon_j(x_{j+1/2})-\rho\varepsilon_j(x_j)}{x_{j+1/2}-x_j} \end{bmatrix}, \\
\boldsymbol{W}_r = \int \boldsymbol{\Psi}_\alpha g_r \mathrm{d}\varepsilon = \begin{bmatrix} \rho_{j+1}(x_{j+1/2}) \\ \rho_{j+1} U_{j+1}(x_{j+1/2}) \\ \rho_{j+1}\varepsilon_{j+1}(x_{j+1/2}) \end{bmatrix},
\end{cases}
\tag{3-171}
$$

$$
\boldsymbol{F}_{W_r} = \int \boldsymbol{\Psi}_\alpha g_r \mathrm{d}\varepsilon = \begin{bmatrix} \dfrac{\rho_{j+1}(x_{j+1})-\rho_{j+1}(x_{j+1/2})}{x_{j+1}-x_{j+1/2}} \\ \dfrac{\rho_{j+1}U_{j+1}(x_{j+1})-\rho_{j+1}U_{j+1}(x_{j+1/2})}{x_{j+1}-x_{j+1/2}} \\ \dfrac{\rho_{j+1}\varepsilon_{j+1}(x_{j+1})-\rho_{j+1}\varepsilon_{j+1}(x_{j+1/2})}{x_{j+1}-x_{j+1/2}} \end{bmatrix}. \tag{3-172}
$$

考虑到 $g = \rho\left(\dfrac{\lambda}{\pi}\right)^{\frac{k+1}{2}} \mathrm{e}^{-\lambda[(u-U)^2+\xi^2]}$,可以得到

$$
\frac{1}{\rho_r}\begin{bmatrix} \dfrac{\rho_{j+1}(x_{j+1})-\rho_{j+1}(x_{j+1/2})}{x_{j+1}-x_{j+1/2}} \\ \dfrac{\rho_{j+1}U_{j+1}(x_{j+1})-\rho_{j+1}U_{j+1}(x_{j+1/2})}{x_{j+1}-x_{j+1/2}} \\ \dfrac{\rho_{j+1}\varepsilon_{j+1}(x_{j+1})-\rho_{j+1}\varepsilon_{j+1}(x_{j+1/2})}{x_{j+1}-x_{j+1/2}} \end{bmatrix} = \frac{1}{\rho_r}\begin{bmatrix} \left(\dfrac{\partial\rho}{\partial x}\right)^r \\ \left(\dfrac{\partial\rho U}{\partial x}\right)^r \\ \left(\dfrac{\partial\rho\varepsilon}{\partial x}\right)^r \end{bmatrix} = \boldsymbol{M}^r_{\alpha\beta}\begin{bmatrix} a_{1r} \\ a_{2r} \\ a_{3r} \end{bmatrix}, \tag{3-173}
$$

其中

$$
\boldsymbol{M}^r_{\alpha\beta} = \begin{bmatrix} 1 & U_r & \dfrac{1}{2}\left(U_r^2+\dfrac{k+1}{2\lambda_r}\right) \\ U_r & U_r^2+\dfrac{1}{2\lambda_r} & \dfrac{1}{2}\left(U_r^3+\dfrac{(k+3)U_r}{2\lambda_r}\right) \\ \dfrac{1}{2}\left(U_r^2+\dfrac{k+1}{2\lambda_r}\right) & \dfrac{1}{2}\left(U_r^3+\dfrac{(k+3)U_r}{2\lambda_r}\right) & \dfrac{1}{4}\left(U_r^4+\dfrac{(k+3)U_r^2}{\lambda_r}+\dfrac{K^2+4k+3}{4\lambda_r^2}\right) \end{bmatrix}, \tag{3-174}
$$

以及

$$
\begin{cases}
a_{3r} = \dfrac{4\lambda_r^2}{k+1}\left[2\left(\dfrac{\partial \varepsilon}{\partial x}\right)^r - 2U_r\left(\dfrac{\partial U}{\partial x}\right)^r\right], \\
a_{2r} = 2\lambda_r\left[\left(\dfrac{\partial U}{\partial x}\right)^r - \dfrac{U_r}{2\lambda_r}a_{3r}\right], \\
a_{1r} = \dfrac{1}{\rho_r}\left(\dfrac{\partial \rho}{\partial x}\right)^r - U_r a_{2r} - \left(\dfrac{U_r^2}{2} + \dfrac{k+1}{4\lambda_r}\right)a_{3r}, \\
\left(\dfrac{\partial U}{\partial x}\right)^r = \dfrac{1}{\rho_r}\left[\left(\dfrac{\partial \rho \varepsilon}{\partial x}\right)^r - U_r\left(\dfrac{\partial \rho}{\partial x}\right)^r\right], \\
\left(\dfrac{\partial \varepsilon}{\partial x}\right)^r = \dfrac{1}{\rho_r}\left[\left(\dfrac{\partial \rho \varepsilon}{\partial x}\right)^r - \dfrac{1}{2}\left(U_r^2 + \dfrac{k+1}{2\lambda_r}\right)\left(\dfrac{\partial \rho}{\partial x}\right)^r\right].
\end{cases}
\tag{3-175}
$$

对于左侧,有与(3-171)—(3-175)式相同的公式,只要将 r 改为 l 就可以了.

在 g_l、g_r、a_l、a_r 确定以后,再根据(3-168)式可以得到 f_0,然后代入(3-165)式,可以得到 $j+1/2$ 处的守恒量通量.由此确定该处的守恒量.具体说是在 $j+1/2$ 处即形成平衡态的分布 g_0.确定 g_0 的公式为(g_0 为 $j+1/2$ 处的平衡态分布)

$$
\int g_0 \Psi_\alpha \mathrm{d}\varepsilon = \int_{u>0}\int \Psi_\alpha g_l \mathrm{d}\varepsilon + \int_{u<0}\int \Psi_\alpha g_r \mathrm{d}\varepsilon. \tag{3-176}
$$

引入记号

$$
\rho_l\langle \cdot \rangle_{>0} = \int_{u>0}\int (\cdot) g_l \mathrm{d}\varepsilon
$$

$$
\rho_r\langle \cdot \rangle_{<0} = \int_{u<0}\int (\cdot) g_l \mathrm{d}\varepsilon
$$

其中"·"表示为某一量.代入(3-176)式得

$$
\begin{bmatrix} \rho_{j+1/2} \\ \rho_{j+1/2}U_{j+1/2} \\ \rho_{j+1/2}\varepsilon_{j+1/2} \end{bmatrix} = \begin{bmatrix} \rho_0 \\ \rho_0 U_0 \\ \rho_0\varepsilon_0 \end{bmatrix} = \begin{bmatrix} \rho_l\langle u^0\rangle_{>0} + \rho_r\langle u^0\rangle_{<0} \\ \rho_l\langle u^1\rangle_{>0} + \rho_r\langle u^1\rangle_{<0} \\ \dfrac{1}{2}[\rho_l\langle u^2+\xi^2\rangle_{>0} + \rho_r\langle u^2+\xi^2\rangle_{<0}] \end{bmatrix}, \tag{3-177}
$$

其中 u^0 为 u 的零次方,故 $u^0 = 1$.(3-177)式中右端项内的各个量可以由附录 B 中的公式计算,所以 ρ_0、$\rho_0 U_0$、$\rho_0\varepsilon_0$ 可以由(3-177)式确定.有了 g_0(即 $j+1/2$ 处的初始平衡态的分布)后再用(3-169)就可以确定 g,而其中的 $\bar{a}_r$、$\bar{a}_l$ 可以由下列关系得到(确定方法与上面讨论是类似的):

$$
\frac{1}{\rho_0}\begin{bmatrix} \dfrac{\rho_{j+1}(x_{j+1}) - \rho_{j+1/2}}{x_{j+1} - x_{j+1/2}} \\ \dfrac{\rho_{j+1}U_{j+1}(x_{j+1}) - \rho_{j+1/2}U_{j+1/2}}{x_{j+1} - x_{j+1/2}} \\ \dfrac{\rho_{j+1}\varepsilon_{j+1}(x_{j+1}) - \rho_{j+\frac{1}{2}}\varepsilon_{j+\frac{1}{2}}}{x_{j+1} - x_{j+1/2}} \end{bmatrix} = M_{\alpha\beta}^0 \begin{bmatrix} \bar{a}_{1r} \\ \bar{a}_{2r} \\ \bar{a}_{3r} \end{bmatrix}, \tag{3-178}
$$

$$
M^0_{\alpha\beta}=\begin{bmatrix}
1 & U_{j+1/2} & \frac{1}{2}\left(U^2_{j+1/2}+\frac{k+1}{2\lambda_{j+1/2}}\right)\\
U_{j+1/2} & U^2_{j+1/2}+\frac{k}{2\lambda_{j+1/2}} & \frac{1}{2}\left(U^3_{j+1/2}+\frac{(k+3)U_{j+1/2}}{\lambda_{j+1/2}}\right)\\
\frac{1}{2}\left(U^2_{j+1/2}+\frac{k+1}{2\lambda_{j+1/2}}\right) & \frac{1}{2}\left(U^3_{j+1/2}+\frac{(k+3)U_{j+1/2}}{\lambda_{j+1/2}}\right) & \frac{1}{4}\left(U^4_{j+1/2}+\frac{(k+3)U^2_{j+1/2}}{\lambda_{j+1/2}}+\frac{k^2+4+3}{4\lambda^2_{j+1/2}}\right)
\end{bmatrix}.
\tag{3-179}
$$

而$(\bar{a}_{1r},\bar{a}_{2r},\bar{a}_{3r})$ 的计算与(3-175)式相同. 对左侧,$j-1/2$ 处的公式与(3-177)—(3-179)式相同,只需把 r 改为 l,$j+1/2$ 改为 $j-1/2$.这样得到的是 $t=0$ 的初始状态.有了上面的 g、f_0,代入 BGK 模型的通解(3-151)式有

$$
f[x_{j+1/2},t,u,\xi]=\frac{1}{\tau}\int_0^t g(x',t',u,\xi)\mathrm{e}^{-(t-t')/\tau}\mathrm{d}t'+\mathrm{e}^{-t/\tau}f_0(x_{j+1/2}-ut),
\tag{3-180}
$$

于是有

$$
\begin{aligned}
f[x_{j+1/2},t,u,\xi]=&(1-\mathrm{e}^{-t/\tau})g_0+(\tau(-1+\mathrm{e}^{-t/\tau})+t\mathrm{e}^{-t/\tau})\cdot\\
&(\bar{a}_lH(u)+\bar{a}_r(1-H(u)))ug_0+\tau\left(\frac{t}{\tau}-1+\mathrm{e}^{-t/\tau}\right)\bar{A}g_0+\\
&\mathrm{e}^{-t/\tau}((1-uta_l)H(u)g_l+(1-uta_r)(1-H(u))g_r).
\end{aligned}
\tag{3-181}
$$

上式中 $\bar{A}$ 是不知道的,为此可以用相容条件$\int_0^{\Delta t}\int(g-f)\Psi_\alpha\mathrm{d}\epsilon\mathrm{d}t=0$确定,该式可以在 g、f 代入后得到

$$
\begin{aligned}
\overline{M}^0_{\alpha\beta}\bar{A}=\frac{1}{\rho_0}\int\{&\gamma_1g_0+\gamma_2u[\bar{a}_lH(u)+\bar{a}_r(1-H(u))]g_0+\\
&\gamma_3[H(u)g_l+(1-H(u))g_r]+\\
&\gamma_4u[a_lH(u)g_l+a_r(1-H(u))g_r]\}\Psi_\alpha\mathrm{d}\epsilon,
\end{aligned}
\tag{3-182}
$$

其中

$$
\begin{aligned}
\gamma_0&=\Delta t-\tau(1-\mathrm{e}^{-\Delta t/\tau}),\\
\gamma_1&=-(1-\mathrm{e}^{-\Delta t/\tau})/\gamma_0,\\
\gamma_2&=[-\Delta t+2\tau(1-\mathrm{e}^{-\Delta t/\tau})-\Delta t\mathrm{e}^{-\Delta t/\tau}]/\gamma_0,\\
\gamma_3&=(1-\mathrm{e}^{-\Delta t/\tau})/\gamma_0,\\
\gamma_4&=[\Delta t\mathrm{e}^{-\Delta t/\tau}-\tau(1-\mathrm{e}^{-\Delta t/\tau})]/\gamma_0.
\end{aligned}
\tag{3-183}
$$

$\overline{M}^0_{\alpha\beta}$ 与(3-179)同取 $x_{j+1/2}$ 处的 ρ、ρU、$\rho\epsilon$ 值.有了 f,则

$$
\begin{bmatrix}F_\rho\\F_{\rho U}\\F_{\rho\epsilon}\end{bmatrix}_{j+1/2}=\int u\begin{bmatrix}1\\u\\\frac{1}{2}(u^2+\xi^2)\end{bmatrix}f(x_{j+1/2},t,u,\xi)\mathrm{d}\epsilon.
\tag{3-184}
$$

再代入公式(3-154)式,就可以求得守恒量的时间变化,即 $n+1$ 时刻的守恒量的值.

以上的格式可以推广到多维的情况，二维时的 N-S 方程为

$$\begin{bmatrix}\rho\\ \rho U\\ \rho V\\ \rho\epsilon\end{bmatrix}_t+\begin{bmatrix}\rho U\\ \rho U^2+p\\ \rho UV\\ (\rho\epsilon+p)U\end{bmatrix}_x+\begin{bmatrix}\rho V\\ \rho UV\\ \rho V^2+p\\ (\rho\epsilon+p)U\end{bmatrix}_y=\begin{bmatrix}0\\ S_{1x}\\ S_{2x}\\ S_{3x}\end{bmatrix}_x+\begin{bmatrix}0\\ S_{1y}\\ S_{2y}\\ S_{3y}\end{bmatrix}_y.$$

其中

$$S_{1x}=\tau p\left[2\frac{\partial U}{\partial x}-\frac{2}{5}\left(\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}\right)\right],$$

$$S_{1y}=\tau p\left[\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right],$$

$$S_{2x}=\tau p\left[\frac{\partial V}{\partial x}+\frac{\partial U}{\partial y}\right],$$

$$S_{2y}=\tau p\left[2\frac{\partial V}{\partial y}-\frac{2}{5}\left(\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}\right)\right],$$

$$S_{3x}=\tau p\left[2U\frac{\partial U}{\partial x}+V\left(\frac{\partial V}{\partial x}+\frac{\partial U}{\partial y}\right)-\frac{2}{5}U\left(\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}\right)+\frac{7}{4}\frac{\partial}{\partial x}\left(\frac{1}{\lambda}\right)\right],$$

$$S_{3y}=\tau p\left[U\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right)+2V\frac{\partial V}{\partial y}-\frac{2}{5}V\left(\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}\right)+\frac{7}{4}\frac{\partial}{\partial x}\left(\frac{1}{\lambda}\right)\right].$$

三维有类似公式(注意 $\mu=\tau p$). 这时 BGK 格式为

$$W_{ij}(x,y)=W_{ij}(x_i,y_j)+L_x(W,W)(x-x_{i,j})+L_y(W,W)(y-y_{i,j}),$$

其中 W_{ij} 是$(x_{i-1/2},x_{i+1/2})(y_{j-1/2},y_{j+1/2})$区间内的守恒量值，$L$ 为限制器，

$$L_x(W,W)=L_x\left(\frac{W_{i+1,j}-W_{ij}}{x_{i+1}-x_i},\frac{W_{ij}-W_{i-1,j}}{x_i-x_{i-1}}\right),$$

$$L_y(W,W)=L_y\left(\frac{W_{i,j+1}-W_{ij}}{y_{j+1}-y_j},\frac{W_{ij}-W_{i,j-1}}{y_j-y_{j-1}}\right).$$

在计算 W^{n+1} 时可以用时间分裂的方法，即

$$W_t+F(W)_x=S_x,\quad W_t+F(W)_y=S_y.$$

如果不用时间分裂则可用以下的公式，比如计算 $x_{i+1/2},y_j$ 处的通量时，先要计算该处的分布函数，这时

$$f_0(x,y,0)=\begin{cases}g_l(1+a_lx+b_ly), & x<0\\ g_r(1+a_rx+b_ry), & x>0\end{cases}$$

$$=g_l(1+a_lx+b_ly)(1-H(x))+g_r(1+a_rx+b_ry)(1-H(x)),$$

其坐标原点迁到了$(x_{i+1/2},y_j)$处，该处为 $x=0,y=0$，分布函数则为

$$g_l=\rho_l\left(\frac{\lambda_l}{\pi}\right)^{\frac{k+2}{2}}e^{-\lambda_l[(u-U_l)^2+(v-V_l)^2+\xi^2]},$$

$$g_r=\rho_r\left(\frac{\lambda_r}{\pi}\right)^{\frac{k+2}{2}}e^{-\lambda_r[(u-U_r)^2+(v-V_r)^2+\xi^2]},$$

其中

$$a_l = a_{1l} + a_{2l}u + a_{3l}v + a_{4l}\frac{1}{2}(u^2 + v^2 + \xi^2),$$

$$a_r = a_{1r} + a_{2r}u + a_{3r}v + a_{4r}\frac{1}{2}(u^2 + v^2 + \xi^2),$$

$$b_l = b_{1l} - b_{2l}u + b_{3l}v + b_{4l}\frac{1}{2}(u^2 + v^2 + \xi^2),$$

$$b_r = b_{1r} + b_{2r}u + b_{3r}v + b_{4r}\frac{1}{2}(u^2 + v^2 + \xi^2).$$

在交接面上
$$\overline{W} = \int \Psi_a f_0 \mathrm{d}\varepsilon,$$

其中
$$\Psi_a = \left(1, u, v, \frac{1}{2}(u^2 + v^2 + \xi^2)\right)^{\mathrm{T}}.$$

同时

$$\overline{W} = W(x_{j+1/2}, y_j) = W_{ij} + L_x(x - x_{ij}) + L_y(y - y_j),$$

其中

$$\frac{1}{\rho_l}L_x = \frac{1}{\rho_l}\begin{bmatrix}(\partial\rho/\partial x)^l\\(\partial\rho U/\partial x)^l\\(\partial\rho V/\partial x)^l\\(\partial\rho\varepsilon/\partial x)^l\end{bmatrix} = \frac{1}{\rho_l}\int \Psi_a a_l g_l \mathrm{d}\epsilon = M_l\begin{bmatrix}a_{1l}\\a_{2l}\\a_{3l}\\a_{4l}\end{bmatrix},$$

$$\frac{1}{\rho_l}L_y = \frac{1}{\rho_l}\begin{bmatrix}(\partial\rho/\partial y)^l\\(\partial\rho U/\partial y)^l\\(\partial\rho V/\partial y)^l\\(\partial\rho\varepsilon/\partial y)^l\end{bmatrix} = \frac{1}{\rho_l}\int \Psi_a b_l g_l \mathrm{d}\epsilon = M_l\begin{bmatrix}b_{1l}\\b_{2l}\\b_{3l}\\b_{4l}\end{bmatrix}.$$

由 $M_l = \left(\frac{1}{\rho_l}\int \Psi_a \Psi_\beta\, g_l \mathrm{d}\epsilon\right)$ 得

$$M_l = \begin{bmatrix}1 & U_l & V_l & B_1\\ U_l & U_l^2 + \frac{1}{2\lambda_l} & U_lV_l & B_2\\ V_l & U_lV_l & V_l^2 + \frac{1}{2\lambda_l} & B_3\\ B_1 & B_2 & B_3 & B_4\end{bmatrix},$$

其中
$$B_1 = \frac{1}{2}\left(U_l^2 + V_l^2 + \frac{k+2}{2\lambda_l}\right),$$

$$B_2 = \frac{1}{2}\left(U_l^3 + V_l^2U_l + \frac{(k+4)U_l}{2\lambda_l}\right),$$

$$B_3 = \frac{1}{2}\left(V_l^3 + U_l^2V_l + \frac{(k+4)U_l}{2\lambda_l}\right),$$

$$B_4 = \frac{1}{4}\left[(U_l^2 + V_l^2)^2 + \frac{(k+4)(U_l^2 + V_l^2)}{\lambda_l} + \frac{k^2 + 6k + 8}{4\lambda_l^2}\right],$$

在右侧也有类似的公式. 在确定 a_r、a_0、b_r、b_l 后即得到两单元交接面上的f_0和

前面一维时的办法一样,可以确定相应的 g_0、f,最后计算通过交接面的通量.

综上所述,计算步骤如下:

(1) 初值再构.若网格内物理量为常数,则没有重构的问题;若为线性变化,则用限制器计算交接面上的物理量,然后由此确定交接面左右侧的平衡态分布函数.

(2) 通量计算.交接面处的通量计算是这样确定的:有了左右侧的 g_l、g_r,即得到 f_0,再用相容条件可以确定 g_0,进而由 BGK-玻耳兹曼方程通解确定 f,交接面处的 f 也就确定了,计算其相应的通量.

(3) 计算单元内物理量.由于各交接面的通量已确定,就可以由有限体积法确定这些物理量.

不难看出,这三步对于所有格式都是正确的,不同之处只在第二步,即通量的确定.在TVD等格式中,一般用黎曼解得到的通量来计算,或将其改进或修正,而这里采用的是玻耳兹曼方程求解过程来确定.应用表明该方法更加稳健,有更好的保证性,有满足熵条件等优点.

3.7 时空守恒格式

迄今为止,人们采用的守恒格式实际上都只是在空间守恒,而在时间方向并不守恒.为了提高计算精度,也应当考虑时间方向上的守恒.为此,Chang 提出了时空守恒元和解元的新概念,发展了时空守恒格式.在此后,张增产对 Chang 的方法进行了改进,改变了原格式的构造方法,得到一种新的格式.它也是时空守恒的,不仅保留了原方法的全部优点,而且格式构造更加简单,并且更加容易推广到高阶格式和多维格式.下面将介绍改进后的时空守恒格式.

在时空守恒格式中,时间和空间是统一处理的.从守恒方程出发,通过设立守恒元和解元,使格式在局部和全局都保持守恒.在该格式中将流场的变量及其空间方向的偏导数都视为独立变量,同时进行求解,这样一方面可以提高格式精度,另一方面也便于更精确的处理边界条件.由于篇幅的限制,本节只讨论一维尤拉方程的时空守恒格式.欧拉方程可以写成

$$\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}=0, \tag{3-185}$$

其中 $\boldsymbol{U}=(\rho,\rho u,\rho e))^{\mathrm{T}}$,$\boldsymbol{F}=(\rho u,\rho u^2+P,\rho u(e+P))^{\mathrm{T}}$.若设 $x_1=t$,$x_2=x$ 表示欧几里德空间 E_2 中的两个坐标,则利用高斯散度公式,(3-185) 式积分得

$$\oint_{\partial\Omega}\boldsymbol{h}_i\cdot \mathrm{d}\boldsymbol{s}=0, \tag{3-186}$$

其中 $\boldsymbol{h}_i=(F_i,U_i)$,$(i=1,2,3)$,$\mathrm{d}\boldsymbol{s}$ 以逆时针方向为正方向,$\boldsymbol{x}$ 为边界元向量,Ω 为 x_1-x_2 平面上的一个区域,$\partial\Omega$ 为其边界.在积分域为网格时,$\partial\Omega$ 就是网格的边界.

在时空 x-t 平面上如图 3-10 地划分网格,其中 o、x 是不同时间层上的网格点,他们在 x、t 轴上的投影是交替的.

图中 $BCEF$ 称作守恒元,它布满全部 x 空间,积分在其上进行.而 $BDFG$ 称作解元,解在该区域可以用简单函数逼近.

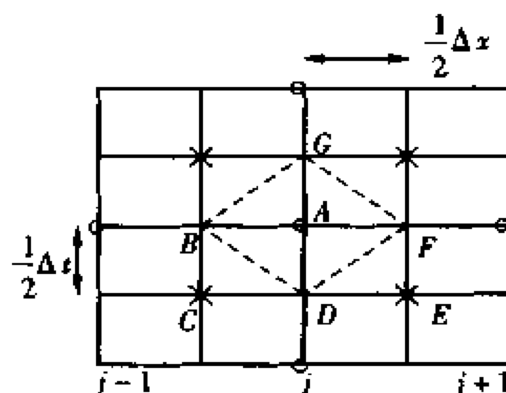


图 3-10

这里用 SE 表示解元(solution element) 和 CE(conservation element) 表示守恒元. 其中用 u^*、f^* 来逼近函数 u、f 可以有不同的形式和精度，于是得到的格式精度将是不同的. 最简单的方法是泰勒展开，在下面将采用这样的方法来构造格式. 如采用一阶的泰勒式来进行逼近，则 $SE(j,n)$(j,n 表示 A 点的位置，即 x_j,t_n) 内任一点的函数

$$\begin{cases} u^*(x,t;j,n) = (u^*)_j^n + (u_x^*)_j^n \cdot \\ \quad (x - x_j) + (u_t^*)_j^n(t - t_n), \\ f^*(x,t;j,n) = (f^*)_j^n + (f_x^*)_j^n \cdot \\ \quad (x - x_j) + (f_t^*)_j^n(t - t_n), \end{cases} \tag{3-187}$$

其中 u、f 是 $\boldsymbol{U}$、$\boldsymbol{F}$ 的分量，应记作 u_m、f_m($m = 1,2,3$)，为简单起见略去了 m 下标. * 表示解元内的点的函数值. 为简单起见，分别简写为 $u(x,t;j,n)$ 和 $f(x,t;j,n)$，组合记作

$$\boldsymbol{h}(x,t;j,n) = (f(x,t;j,n), u(x,t;j,n)). \tag{3-188}$$

方程(3-185) 在(j,n) 点上可以写作

$$(u_{m,t})_j^n = -(f_{m,x})_j^n, \quad m = 1,2,3. \tag{3-189}$$

由于 $f_x = \sum_{m=1}^{3}\frac{\partial f}{\partial u_m}u_{m,x}$，所以最后还是归结为 u、$u_{m,x}$ 是要求解的量. 将(3-187) 和 (3-188) 式代入(3-186) 式，其中 Ω 为守恒元 CE(即 $CEFB$ 单元). 可以得到

$$(u_m)_j^n = [(u_m)_{j-1/2}^{n-1/2} + (u_m)_{j+1/2}^{n-1/2} + (s_m)_{j-1/2}^{n-1/2} + (s_m)_{j+1/2}^{n-1/2}]/2, \tag{3-190}$$

其中

$$s_m = \frac{\Delta x}{4}u_{m,x} + \frac{\Delta t}{\Delta x}\left(f_m + \frac{\Delta t}{4}f_{m,t}\right), \quad m = 1,2,3. \tag{3-191}$$

另外由图中 B、F 点处 u_m^*、f_m^* 的连续性要求，则应有

$$\begin{cases} B\text{ 点}: (u_m)_j^n - \frac{\Delta x}{2}(u_{m,x})_j^n = (u_m)_{j-1/2}^{n-1/2} + \frac{\Delta t}{2}(u_{m,t})_{j-1/2}^{n-1/2} \overset{\text{def}}{=\!=} (\tilde{u}_m)_{j-1/2}^n, \\ F\text{ 点}: (u_m)_j^n + \frac{\Delta x}{2}(u_{m,x})_j^n = (u_m)_{j+1/2}^{n-1/2} + \frac{\Delta t}{2}(u_{m,t})_{j+1/2}^{n-1/2} \overset{\text{def}}{=\!=} (\tilde{u}_m)_{j+1/2}^n, \end{cases} \tag{3-192}$$

其中

$$(u_{m,x})_j^n = \frac{(u_{m,x}^+)_j^n + (u_{m,x}^-)_j^n}{2},$$

$$(u_{m,x}^{\pm})_j^n = \pm\frac{(\tilde{u}_{m,x}^{\pm})_{j\pm1/2}^n - (u_m)_j^n}{\Delta x/2}.$$

在有间断的情况下要用加权平均，即相当于引入一个限制器

$$(u_{m,x})_j^n = \frac{|(u_{m,x}^-)|^\alpha (u_{m,x}^+)_j^n + |(u_{m,x}^+)|^\alpha (u_{m,x}^-)_j^n}{|(u_{m,x}^-)_j^n|^\alpha + |(u_{m,x}^+)_j^n|^\alpha}, \tag{3-193}$$

这样(3-190)和(3-193)式就是得到的一维尤拉方程的失控守恒格式.这是显示的格式,保留了原 *CE/SE* 的全部优点,但更加简单.

关于一维对流扩散方程的时空守恒格式,考虑下面一维方程

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}, \tag{3-194}$$

其中 a,μ 为常数,且 $\mu > 0$.这里可以设

$$\boldsymbol{h} = \left(au - \mu\frac{\partial u}{\partial x}, u\right), \tag{3-195}$$

积分得

$$\oint_{\partial\Omega} \boldsymbol{h}\cdot\partial s = 0. \tag{3-196}$$

在解元内也采用和前面一样的泰勒展开,于是有

$$\boldsymbol{h}^*(x,t;j,n) = \left(au^*(x,t;j,n) - \mu\frac{\partial u^*(x,t;j,n)}{\partial x}, u^*(x,t;j,n)\right). \tag{3-197}$$

将(3-196)式代入(3-194)式得

$$(u_t)_j^n = -a(u_x)_j^n, \tag{3-198}$$

由于是一阶泰勒展开,所以没有出现扩散项,所以该项在解元中没有影响.但在二阶泰勒展开中会出现扩散项,而且即使是一阶泰勒展开,当积分对守恒元进行时仍会出现.事实上利用(3-198)式,(3-195)式可以改写为

$$u^*(x,t;j,n) = u_j^n + (u_x)_j^n[(x - x_j) - a(t - t_n)], \tag{3-199}$$

利用(3-188)式在守恒元上积分(即 *BCEF* 元),可写为

$$F_\pm(j,n) = \oint_{(CE)_\pm} \boldsymbol{h}^*\cdot \mathrm{d}s = 0 \tag{3-200}$$

其中$(CE)_+$即 *ADEF*,是 *CE* 中的右半部分,$(CE)_-$即 *ABCD*,是 *CE* 中的左半部分.将(3-199)式代入(3-197)式,再代入上式,计及左右两半的交接线全部落在解元中,所以不用插值.经过积分运算可以得到两个离散方程:

$$\frac{4}{(\Delta x)^2}F_\pm(j,n) = \pm\frac{1}{2}\left[(1 - \nu^2 - \xi)(u_x)_j^n + (1 - \nu^2 - \xi)(u_x)_{j\pm1/2}^{n-1/2}\right] + \frac{2(1\mp\nu)}{\Delta x}(u_j^n - u_{j\pm1/2}^{n-1/2}) = 0, \tag{3-201}$$

其中 $$\nu = \frac{a\Delta t}{\Delta x}, \quad \xi = \frac{4\mu\Delta t}{(\Delta x)^2}.$$

当$(1 - \nu^2 - \xi) \neq 0$时从上式可以解得 u_j^n 和$(u_x)_j^n$为

$$\boldsymbol{q}(j,n) = \boldsymbol{Q}_+\boldsymbol{q}\left(j - \frac{1}{2}, n - \frac{1}{2}\right) + \boldsymbol{Q}_-\boldsymbol{q}\left(j + \frac{1}{2}, n - \frac{1}{2}\right) \tag{3-202}$$

其中

$$q(j,n)=\begin{bmatrix} u_j^n \\ (u_x)_j^n\dfrac{\Delta x}{4}\end{bmatrix},$$

$$Q_+=\frac{1}{2}\begin{bmatrix} 1+\nu & 1-\nu^2-\xi \\ \dfrac{-(1-\nu^2)}{1-\nu^2+\xi} & \dfrac{-(1-\nu)(1-\nu^2+\xi)}{1-\nu^2+\xi}\end{bmatrix},$$

$$Q_-=\frac{1}{2}\begin{bmatrix} 1-\nu & -(1-\nu^2-\xi) \\ \dfrac{1-\nu^2}{1-\nu^2+\xi} & \dfrac{-(1+\nu)(1-\nu^2-\xi)}{1-\nu^2+\xi}\end{bmatrix}.$$

这是一个半层显式格式，连续用两次即可得到通常形式的显式格式

$$q(j,n+1)=Q_+^2\,q(j-1,n)+(Q_+Q_-+Q_-Q_+)q(j,n)+Q_-^2\,q(j,n).$$

上面的计算中必须注意 $1-\nu^2+\xi\neq 0$！(3-212) 式又称 a-μ 格式.

在这个时空格式中可以看到将时间和空间完全等同起来，而且 u、u_x 作为两个独立变量来处理. 格式的网格结构十分简单，某一层的参数值只与前半层上的两个点值相关，而且还不会因泰勒展开式的精度的提高而增加点数. 在上面的计算中，实际上用到在解元中满足微分方程，而在守恒律中满足积分方程. 由于时间方向上网格交替进行，所以有

$$q(j,n+1)\to q(j,n)\quad(\Delta t\to 0,a,\mu,\Delta x=\text{const}),$$

这是因为在 a、μ、Δx 保持不变时，$\Delta t\to 0$ 导致

$$\begin{cases} Q_+^2\to 0, \\ Q_-^2\to 0,\quad (Q_+Q_-+Q_-Q_+)\to 1.\end{cases}$$

这些性质显然是其它的格式所不具备的.

以上格式的稳定条件为 $\nu^2\leq 1$. 进一步分析(读者可自行证明)表明，$\mu=0$ 时其放大因子与蛙跳式的放大因子相同，故 $\nu<1$ 时为中性稳定. 而当 $a=0$ 时，放大因子与 Difort-Frankel 格式的放大因子相同，格式为无条件稳定.

这里的稳定条件 $\nu^2\leq 1$，即 $\nu<1$，要求 Δx、Δt 成正比，所以这个稳定条件是十分严格的. 另一个重要的问题是 $\mu=0$ 时是中性稳定的，其中 a 是常数，也就是说方程是线性的. 在此条件下，如方程是非线性的，格式就会不稳定. 为此需要对方法进行改进. 下面讨论的 a-ϵ 格式就是这样发展起来的.

a-ϵ 格式采用的方法是追加人工粘性项中取 $\mu=0$. 推导过程与前类似，只是 (3-201) 式中原来设 $F_\pm(j,n)=0$，现改为

$$F_\pm(j,n)=\pm\frac{\epsilon(1-\nu^2)\Delta x^2}{4}(\mathrm{d}u_x)_j^n,\tag{3-203}$$

其中 ϵ 是一个粘性项控制参数(与流场变量无关). 由于 $F_\pm$ 是 $O(\Delta x^2)$ 量级，所以不会降低原格式的精度，而(3-203) 式中

$$(\mathrm{d}u_x)_j^n=\frac{1}{2}[(u_x)_{j-1/2}^{n-1/2}+(u_x)_{j+1/2}^{n-1/2}]-(u_{j+1/2}^{n-1/2}-u_{j-1/2}^{n-1/2})/\Delta x.\tag{3-204}$$

将(3-192) 式右端项 O 用(3-196) 式的右端项代替，则得到

$$q(j,n) = M_+ q\left(j - \frac{1}{2}, n - \frac{1}{2}\right) + M_- q\left(j + \frac{1}{2}, n - \frac{1}{2}\right),$$
$$1 - v^2 + \xi \neq 0 \qquad (3\text{-}205)$$

和

$$q(j,n) = M_+ q\left(j - \frac{1}{2}, n - \frac{1}{2}\right) + M_- q\left(j + \frac{1}{2}, n - \frac{1}{2}\right),$$
$$1 - v^2 + \xi \neq 0, \qquad (3\text{-}206)$$

其中

$$M_+ = \frac{1}{2}\begin{bmatrix} 1+v & 1-v^2 \\ \varepsilon - 1 & 2\varepsilon - 1 + v \end{bmatrix}, \quad M_- = \frac{1}{2}\begin{bmatrix} 1-v & -(1-v^2) \\ 1-\varepsilon & 2\varepsilon - 1 - v \end{bmatrix}.$$

格式(3-205) 和(3-206) 就是 a-ε 格式.(3-205) 式也可以写得更详细一点,为

$$\begin{cases} u_j^n = \dfrac{1}{2}[(1+v)u_{j-1/2}^{n-1/2} + (1-v)u_{j+1/2}^{n-1/2}] + \dfrac{\Delta x(1-v^2)}{8}[(u_x)_{j+1/2}^{n-1/2} + (u_x)_{j-1/2}^{n-1/2}], \\ (u_x)_j^n = \dfrac{u_{j+1/2}^n - u_{j-1/2}^n}{\Delta x} + (2\varepsilon - 1)(\mathrm{d}u_x)_j^n, \\ u_{j\pm 1/2}^n = u_{j\pm 1/2}^{n-1/2} + \dfrac{\Delta t}{2}(u_t)_{j\pm 1/2}^{n-1/2}. \end{cases}$$
$$(3\text{-}207)$$

从上式可以看出,尽管加入了人工粘性项,在每一个守恒元的左右,各部分中无法再保持通量守恒,但在守恒元 CE 中仍然是守恒的,即

$$F(j,n) = F_+(j,n) + F_-(j,n) = 0. \qquad (3\text{-}208)$$

加入人工粘性项不会降低计算精度,而且可以用来解非线性问题了.该格式的稳定条件为

$$0 \leq \varepsilon \leq 1, \quad v^2 < 1. \qquad (3\text{-}209)$$

关于这些格式在欧拉和纳维-斯托克斯格式中的推广,可以参考有关的文献,在这里不作详细介绍.

3.8 多维问题和 N-S 方程求解

以上讨论的许多格式多数只是针对一维的情况,对于实际应用而言,多维问题更为有用.在多维的情况下,气体动力学方程具有如下的形式:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} + \frac{\partial \boldsymbol{g}}{\partial y} + \frac{\partial \boldsymbol{h}}{\partial z} = 0. \qquad (3\text{-}210)$$

解多维问题有两种办法,一种用时间分裂,即分成不同时间步,如分解为三个方程

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} = 0, \quad \frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{g}}{\partial y} = 0, \quad \frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{h}}{\partial z} = 0. \qquad (3\text{-}211)$$

各自分别用一维格式,解的一般形式为

$$u^{n+1/3} = \mathrm{L}_x u^n, \quad u^{n+2/3} = \mathrm{L}_y u^{n+1/3}, \quad u^{n+1} = \mathrm{L}_z u^{n+2/3},$$
$$u^{n+1} = \mathrm{L}_x \mathrm{L}_y L_z u^n. \qquad (3\text{-}212)$$

为保证计算在各个方向上是等价的，方程中的三个方向的算子的顺序应进行轮换.

另一种方法是将三个方向上的一维格式写在一起，即

$$u_{ijk}^{n+1} = u_{ijk}^{n} + L_{h_x}u_i^n + L_{h_y}u_j^n + L_{h_z}u_k^n, \tag{3-213}$$

这样就可以用来解多维问题.显然多维问题的精度、TVD性质和熵条件都是很模糊的.只能由计算结果来说明问题.

在以上讨论中，只考虑无粘流动的情况，实际上流体是有粘性的，这时需要解纳维-斯托克斯方程.在一维时、方程可写为如下的形式：

$$\frac{\partial u}{\partial t} + \frac{\partial(f + f_v)}{\partial x} = 0, \tag{3-214}$$

其中 f_v 为粘性项，它具有一阶导数的形式，它相应的差分格式为

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\widetilde{f}_{j+1/2} - \widetilde{f}_{j-1/2}), \quad \widetilde{f}_{j+1/2} = \widehat{f}_{j+1/2} + \widehat{f}_{v,j+1/2}, \tag{3-215}$$

其中 $\widehat{f}_{j+1/2}$ 为无粘部分时，可用以前讨论的方法计算，为有粘部分时，可用中心差分的方法计算.对于多维的推广则与上面的讨论是一样的.

对于复杂的流场，采用无结构网格是很方便的，因此有必要将过去讨论的差分格式推广到无结构网格上来.为简单起见，下面只讨论无粘的情况，有粘情况是不难推广得到的.

下面以二维为例进行讨论.

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = 0, \tag{3-216}$$

其中

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix}, \quad G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{bmatrix}, \tag{3-217}$$

$E = \rho(e + V^2/2)$，e 为内能.不难证明，

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} = \boldsymbol{A}, \boldsymbol{AU} = \boldsymbol{F}, \quad \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} = \boldsymbol{B}, \boldsymbol{BU} = \boldsymbol{G}, \tag{3-218}$$

A、B 的特征为

$$\lambda(\boldsymbol{A}) = (u + c, u, u, u - c), \quad \lambda(\boldsymbol{B}) = (v + c, v, v, v - c), \tag{3-219}$$

c 为声速.一维格式需要把通量进行分裂，令

$$\boldsymbol{F}_k = \boldsymbol{F}k_1 + \boldsymbol{G}k_2, \quad \boldsymbol{P} = \boldsymbol{A}k_1 + \boldsymbol{B}k_2, \quad \boldsymbol{A} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}, \quad \boldsymbol{B} = \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}},$$

$$V_n = uk_1 + vk_2, \quad V_\tau = vk_1 - uk_2, \quad V^2 = u^2 + v^2,$$

则 $\boldsymbol{P}$ 可分裂为 $\boldsymbol{P} = \boldsymbol{P}^+ + \boldsymbol{P}^-$，其中

$$\boldsymbol{P} = \boldsymbol{Q\Lambda Q}^{-1},$$

$$\lambda(\boldsymbol{P}) = (V_n + c\sqrt{k_1^2 + k_2^2}, V_n, V_n, V_n - c\sqrt{k_1^2 + k_2^2}),$$

$$\boldsymbol{\Lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4),$$

$$
Q=\begin{bmatrix}
1 & \frac{1}{2c^2} & 0 & \frac{1}{2c^2} \\
V_n & \frac{1}{2c}\left(\frac{V_n}{c}+1\right) & 0 & \frac{1}{2c}\left(\frac{V_n}{c}-1\right) \\
V_\tau & \frac{V_\tau}{2c^2} & \rho & \frac{V_\tau}{2c^2} \\
\frac{V^2}{2} & \frac{1}{2}\left(\frac{V^2}{2c^2}+\frac{V_n}{c}+\frac{1}{\gamma-1}\right) & \rho V_\tau & \frac{1}{2}\left(\frac{V^2}{2c^2}-\frac{V_n}{c}+\frac{1}{\gamma-1}\right)
\end{bmatrix}.
$$

$$
P^{\pm}=Q\Lambda^{\pm}Q^{-1}.
$$

其中

$$
n_1=\frac{k_1}{\sqrt{k_1^2+k_2^2}},\quad n_2=\frac{k_2}{\sqrt{k_1^2+k_2^2}},
$$

$$
\Lambda^{\pm}=\frac{1}{2}(\Lambda\pm|\Lambda|).
$$

然后 F 也可分裂为两个部分 F_k^+ 和 F_k^-，$F_k^{\pm}=P^{\pm}U$.

在无结构网格上采用 TVD 格式，基本网格单元为三角形，流体的速度和自由面的高度定义在网格单元的中心.在单元上对(3-216)式两侧作积分，可得

$$
\iint\limits_{\triangle ABC}\left(\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}\right)\mathrm{d}S=\frac{\partial U}{\partial t}\Delta S_i+\oint\limits_{\triangle ABC}F_n\mathrm{d}l,\quad \iint\limits_{\triangle ABC}H_i\mathrm{d}S\approx H_i\Delta S_i, \tag{3-220}
$$

其中 ΔS_i 为第 i 个单元的面积，在 F_n 中，$k_1=n_1$，$k_2=n_2$.方程(3-220)内，右边沿单元侧边的积分近似为

$$
\oint\limits_{\triangle ABC}F_n\mathrm{d}l=\sum_{kb=1}^{3}F_n^{kb}\Delta l_{kb}=Q_b.
$$

将方程(3-220)右边第一项对时间离散后得差分格式

$$
\frac{U^{n+1}-U^n}{\Delta t}\Delta S_i+Q_{bi}=H_i\Delta S_i. \tag{3-221}
$$

一般说来两个单元的公共边的两侧有不同的 F_n，故实际通过这一公共边的通量可由黎曼解求得.通量计算如下(这里用 NND 格式)：

$$
\begin{cases}
F_{nb}=F_{nl}^{+}+F_{nr}^{-}, \\
F_{nl}^{+}=F_{n,kl}^{+}+\overline{\left(\frac{\partial F_n}{\partial x}\right)}_{kl}^{+}\Delta x_{kb,kl}+\overline{\left(\frac{\partial F_n}{\partial y}\right)}_{kl}^{+}\Delta y_{kb,kl}, \\
F_{nl}^{-}=F_{n,kr}^{-}+\overline{\left(\frac{\partial F_n}{\partial x}\right)}_{kr}^{-}\Delta x_{kb,kr}+\overline{\left(\frac{\partial F_n}{\partial y}\right)}_{kr}^{-}\Delta y_{kb,kr},
\end{cases} \tag{3-222}
$$

其中

$$\begin{cases}\overline{\left(\dfrac{\partial F_n}{\partial x}\right)}^{+}_{kl} = \min\bmod\left[\left(\dfrac{\partial F_n^+}{\partial x}\right)_{kl},\left(\dfrac{\partial F_n^+}{\partial x}\right)_{kr}\right],\\ \overline{\left(\dfrac{\partial F_n}{\partial x}\right)}^{-}_{kl} = \min\bmod\left[\left(\dfrac{\partial F_n^-}{\partial_x}\right)_{kl},\left(\dfrac{\partial F_n^-}{\partial x}\right)_{kr}\right],\\ \overline{\left(\dfrac{\partial F_n}{\partial y}\right)}^{+}_{kl} = \min\bmod\left[\left(\dfrac{\partial F_n^+}{\partial y}\right)_{kl},\left(\dfrac{\partial F_n^+}{\partial y}\right)_{kr}\right],\\ \overline{\left(\dfrac{\partial F_n}{\partial y}\right)}^{-}_{kl} = \min\bmod\left[\left(\dfrac{\partial F_n^-}{\partial y}\right)_{kl},\left(\dfrac{\partial F_n^-}{\partial y}\right)_{kr}\right],\end{cases} \tag{3-223}$$

$$\begin{cases}\min\bmod(a,b) = [\mathrm{sign}(a)+\mathrm{sign}(b)]\min(|a|,|b|)/2,\\ \Delta x_{kb,kl} = x_{kb} - x_{kl}, \Delta y_{kb,kl} = y_{kb} - y_{kl},\\ \Delta x_{kb,kr} = x_{kb} - x_{kr}, \Delta y_{kb,kr} = y_{kb} - x_{kr},\end{cases} \tag{3-224}$$

下标 kb 表示 k 单元的一条边，它可以是 $k1,k2$ 或 $k3$，kl 和 kr 表示 kb 边的左、右侧的单元.实际上为了确定 kb 侧上的 $F_n^{\pm}$，不同的方法均可采用，如 Roe 或 Sweby 的方法均可以采用，计算表明它们之间的差别不太大.

在上述的差分格式中，需要计算 F_n 的梯度，该值计算的正确性对计算结果有较大的影响，在此采用最小二乘法计算.在该方法中，用泰勒展开式得

$$\varphi_j = \varphi_i + \frac{\partial\varphi_i}{\partial x}(x_j - x_i) + \frac{\partial\varphi_i}{\partial y}(y_j - y_i) \quad (j = 1,2,3),$$

其中高阶小量略去不计.利用该式可得如下函数：

$$f\left(\frac{\partial\varphi_i}{\partial x},\frac{\partial\varphi_i}{\partial y}\right) = \sum_{j=1}^{3}(\varphi_j - \varphi_i)^2. \tag{3-225}$$

让 f 取极小，可得最佳梯度

$$\begin{cases}\dfrac{\partial\varphi_i}{\partial x} = \dfrac{\sum\limits_{j=1}^{3}\{[(x_j - x_i)r_{22} - (y_j - y_i)r_{12}](\varphi_j - \varphi_i)\}}{r_{11}r_{22} - r_{12}^2},\\ \dfrac{\partial\varphi_i}{\partial y} = \dfrac{\sum\limits_{j=1}^{3}\{[(y_j - y_i)r_{11} - (x_j - x_i)r_{12}](\varphi_j - \varphi_i)\}}{r_{11}r_{22} - r_{12}^2},\end{cases} \tag{3-226}$$

其中

$$r_{11} = \sum_{j=1}^{3}(x_j - x_i)^2, \quad r_{22} = \sum_{j=1}^{3}(y_j - y_i)^2, \quad r_{12} = \sum_{j=1}^{3}(x_j - x_i)(y_j - y_i).$$

4 不可压缩流体流动的差分格式

在可压缩流体流动中，控制方程是纳维-斯托克斯方程，其对流部分在高 Mach 数流动时会出现激波，所以大量格式的重点就放在了激波的捕捉和分辨率上.在不可压缩流场中，尽管控制方程不变，但由于密度不变，连续方程中的 $\partial\rho/\partial t$ 总为零，

因此不能方便地与其它方程联列求解,于是方程求解的重点就在于如何在求解动力学方程时同时保证连续方程得到很好的满足,简言之为 $\mathrm{div}\boldsymbol{V}=0$ 的问题.在这里不存在激波那样的间断,但存在自由面那样的接触间断.

不可压缩流体流动中由于密度是常数,所以在物性与温度无关时能量方程可以在动力学方程求解后单独求解,因此在求解流场时可以暂不考虑能量方程.这时要求解的连续方程和动力学方程可以表示为下面的向量形式:

$$\begin{cases}\nabla\cdot\boldsymbol{V}=0,\\ \dfrac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t}=\dfrac{\partial\boldsymbol{V}}{\partial t}+(\boldsymbol{V}\cdot\nabla)\boldsymbol{V}=-\dfrac{1}{\rho}\nabla p+\nu\Delta\boldsymbol{V}.\end{cases}\tag{4-1}$$

4.1 用投影法解原始变量的 N-S 方程

为了保证速度的散度为零,即满足连续方程,可以用以下的方法来做到.

首先将(4-1)式中动力学方程的压力项略去不计而求解方程

$$\frac{\partial\boldsymbol{V}}{\partial t}+(\boldsymbol{V}\cdot\nabla)\boldsymbol{V}=\nu\Delta\boldsymbol{V},\tag{4-2}$$

该方程在时间方向上离散化得

$$\frac{\boldsymbol{V}^{*}-\boldsymbol{V}^{n}}{\Delta t}+[(\boldsymbol{V}\cdot\nabla)\boldsymbol{V}]^{n}=(\nu\Delta\boldsymbol{V})^{n},\tag{4-3}$$

其中 $\boldsymbol{V}^{*}$ 是一个虚拟的速度场.原动力学方程在时间方向离散后得

$$\frac{\boldsymbol{V}^{n+1}-\boldsymbol{V}^{n}}{\Delta t}+[(\boldsymbol{V}\cdot\nabla)\boldsymbol{V}]^{n}=-\frac{1}{\rho}\nabla p+(\nu\nabla\boldsymbol{V})^{n}.\tag{4-4}$$

由(4-4)式、(4-3)式得

$$\frac{\boldsymbol{V}^{n+1}-\boldsymbol{V}^{*}}{\Delta t}=-\frac{1}{\rho}\nabla p,\tag{4-5}$$

两侧取散度得

$$\frac{\nabla\cdot\boldsymbol{V}^{n+1}-\nabla\cdot\boldsymbol{V}^{*}}{\Delta t}=-\frac{1}{\rho}\nabla^2 p.\tag{4-6}$$

计及连续方程,应当有 $\nabla\cdot\boldsymbol{V}^{n+1}=0$,故得

$$\Delta p=\frac{\rho}{\Delta t}\nabla\cdot\boldsymbol{V}^{*}.\tag{4-7}$$

求解这个压力泊松方程,可得压力 p,再由(4-5)式得知

$$\boldsymbol{V}^{n+1}=\boldsymbol{V}^{*}-\frac{\Delta t}{\rho}\nabla p.\tag{4-8}$$

由这个修正速度的过程得到了满足连续方程的速度场.

为讨论简单起见,首先讨论二维情况,设网格是均匀的,即 Δx、Δy 都是常数. u、v、p 都定义在网格节点 (i,j) 上,则由中心差分方法得

$$\left(\frac{\partial p}{\partial x}\right)_{i,j}=\frac{p_{i+1,j}-p_{i-1,j}}{2\Delta x},\quad\left(\frac{\partial p}{\partial y}\right)_{i,j}=\frac{p_{i,j+1}-p_{i,j-1}}{2\Delta y},$$

于是(4-8)式可改写为

$$\begin{cases} u_{i,j}^{n+1} = u_{i,j}^{*} - \dfrac{\Delta t}{\rho} \dfrac{p_{i+1,j} - p_{i-1,j}}{2\Delta x}, \\ v_{i,j}^{n+1} = v_{i,j}^{*} - \dfrac{\Delta t}{\rho} \dfrac{p_{i,j+1} - p_{i,j-1}}{2\Delta y}, \end{cases} \tag{4-9}$$

可见(i,j)点的速度与该点的压力p无关，而只与周围四个点的压力有关.这样图4-1中压力点可以分解为两套，即○点和×点(图4-1(a))，它们之间没有联系或只靠边界条件松散地联系起来.所以压力的变化可以如图4-1(b)那样呈锯齿状变化，求解过程不收敛而且得到不合理的锯齿解.为克服这个困难，可以采用两种方法：一是用错位网格的方法，即压力p、u、v都不定义在同一点上，而将压力p定义在网格中心，u、v分别定义在左右侧中心和上下侧中心，如图4-2所示.在这种情况下，速度的修正如下：

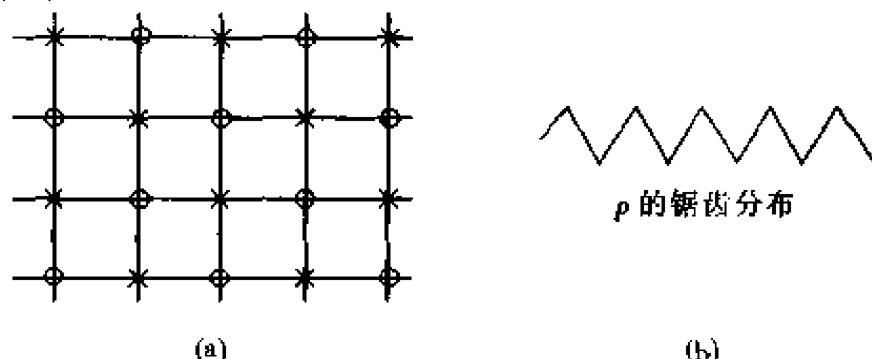


图4-1

$$\begin{cases} u_{i,j}^{n+1} = u_{i,j}^{*} - \dfrac{\Delta t}{\rho} \dfrac{p_{i+1,j} - p_{i,j}}{\Delta x}, \\ v_{i,j}^{n+1} = v_{i,j}^{*} - \dfrac{\Delta t}{\rho} \dfrac{p_{i,j+1} - p_{i,j}}{\Delta y}, \end{cases} \tag{4-10}$$

不难看出，这样得到的V^{n+1}在每个单元上的通量都是零，故$\mathrm{div}\, V^{n+1} = 0$.另一种方法是用动量插值法，即压力$p$修正的是节点间的插值速度，这个方法在4.3节中加以讨论.错位网格法实际上对每一个网格进行了标识，所以也叫MAC法.另一方面，由于这里求解时首先将∇p项略去，而∇p与等压力面是垂直的，所以略去∇p项就相当于动力学方程在压力面上投影，因此人们得到这个名称.

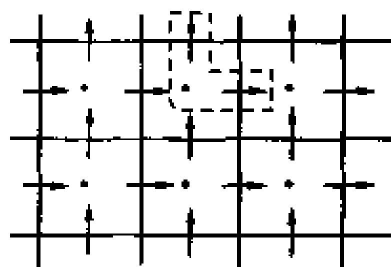


图4-2

下面讨论错位网格的情况.对于一般的空间差分格式需要对网格进行错位，即解u和v所用的网格位置是不同的，如图4-3所示.这时

$$\begin{cases} \dfrac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + a_{i,j}^{n} + \dfrac{p_{i+1,j} - p_{i,j}}{\Delta x} = \dfrac{1}{Re} \nabla^2 u_{i,j}^{n}, \\ \dfrac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta t} + b_{i,j}^{n} + \dfrac{p_{i,j+1} - p_{i,j}}{\Delta x} = \dfrac{1}{Re} \nabla^2 v_{i,j}^{n}, \end{cases} \tag{4-11}$$

其中

$$\begin{cases} a = \dfrac{\partial uu}{\partial x} + \dfrac{\partial uv}{\partial y}, \qquad b = \dfrac{\partial uv}{\partial x} + \dfrac{\partial vv}{\partial y}, \\ a_{i,j}^n = \dfrac{(uu)_{i+1,j} - (uu)_{i-1,j}}{2\Delta x} + \dfrac{(uv)_{i+\frac{1}{2},j+\frac{1}{2}} - (uv)_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y}, \\ b_{i,j}^n = \dfrac{(uv)_{i+\frac{1}{2},j+\frac{1}{2}} - (uv)_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x} + \dfrac{(vv)_{i,j+1} - (vv)_{i,j-1}}{2\Delta y}. \end{cases} \tag{4-12}$$

这里采用的是守恒型中心差分格式，二阶导数也用中心差分格式写出，这样得到的格式的稳定条件为

$$\begin{cases} \dfrac{1}{4}(|u|_{\max} + |v|_{\max})^2 \Delta t \cdot Re \leqslant 1, \\ \dfrac{4\Delta t}{Re \Delta x^2} \leqslant 1. \end{cases} \tag{4-13}$$

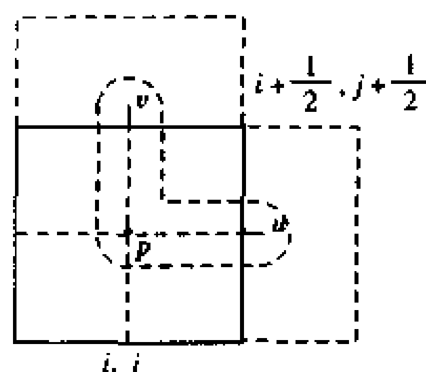


图 4-3

由于求解压力方程时需要边界条件，而实际问题中压力只以一阶导数出现，所以只需给出一个条件. 为求解压力方程需给出数值边界条件. 在壁面上，由于速度已知，又考虑到修正速度的方程为(4-9) 式，因此在壁面上应当取 $\partial p/\partial n = 0$. 另一方面由动力学方程知，应当有

$$\frac{\partial p}{\partial n} = \mu \frac{\partial^2 v_n}{\partial n^2}, \tag{4-14}$$

其中假定壁面为静止，n 是壁面法向，v_n 为速度法向分量. 这个差别是由于计算中得到的压力应当称作数值压力，而与真实的物理压力是不完全相同的. 这是由于数值求解引起的误差，当 μ 比较小，即高 Re 数的情况下，由于(4-14) 式右侧接近于零，所以计算的压力值和真实压力之间误差就很小了.

求解方程(4-2) 的另一种隐式格式为

$$\begin{cases} \dfrac{V^* - V^n}{\Delta t} + \dfrac{3A^n - A^{n-1}}{2} - \dfrac{\nabla^2 V^n}{2Re} = 0, \\ \nabla^2 p = \dfrac{1}{\Delta t} \nabla \cdot V^*, \\ \dfrac{V^{n+1} - V^*}{\Delta t} + \nabla p - \dfrac{\nabla^2 V^{n+1}}{2Re} = 0. \end{cases} \tag{4-15}$$

这个格式又叫 Adams-Bashforth/Crank-Nicolson 格式，其中 $A = (V \cdot \nabla) V$.

关于压力的求解需解一泊松方程，当区域比较简单时可以用快速的直接法求解，具有高精度、高效率；对于复杂形状的流场，则可用多重网格法求解，具体方法可参考有关文献.

以上修正方法是一种显式作法，Spalding 和 Patankar 发展了一种半隐方法(semiimplict method for pressure linked equation，简记 SIMPLE 法). 其基本思想是这样的：

首先把动力学方程写为

$$\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V} \cdot \nabla) \boldsymbol{V} = -\frac{1}{\rho} \nabla p^{*} + \nu \Delta \boldsymbol{V}, \tag{4-16}$$

其中 p^{*} 是一假想的压力初场，比如可以是上一时间层的压力 p^{n}. 利用有限体积法可以将方程(4-2)离散化为

$$\begin{cases} A_e^U U_e = \sum\limits_{nb} A_{nb}^U U_{nb} + B_e^U (p_E - p_P) + q^u, \\ A_n^V V_n = \sum\limits_{nb} A_{nb}^V V_{nb} + B_n^V (p_N - p_P) + q^v, \\ A_t^W W_t = \sum\limits_{nb} A_{nb}^W W_{nb} + B_t^W (p_T - p_P) + q^w, \end{cases} \tag{4-17}$$

其中 A 为系数，p 为(i,j,k)点，nb 表示$(i\pm1,j,k)$、$(i,j\pm1,k)$和$(i,j,k\pm1)$点；B 也是系数，q 是源项；A、B 与几何参数及 n 时间层的值有关，q 与 n 时间层上的值有关；W、E、N、S、B、T 是网格中心点，w、e、n、s、b、t 是中心网格 p 与周围网格交接的侧面. 显然方程(4-16)的离散化方程与(4-17)式相同，只要将式中 U、p 改为 U^{*}、p^{*} 即可. 将所得的方程和原来方程相减，记 $p - p^{*} = p'$，$U - U^{*} = u'$，$V - V^{*} = v'$，$W - W^{*} = w'$，得

$$\begin{cases} A_e^u u'_e = \sum\limits_{nb} A_{nb}^U u'_{nb} + B_e^U (p'_E - p'_P), \\ A_n^V v'_n = \sum\limits_{nb} A_{nb}^V v'_{nb} + B_n^V (p'_N - p'_P), \\ A_t^W w'_t = \sum\limits_{nb} A_{nb}^w w'_{nb} + B_t^W (p'_T - p'_P). \end{cases} \tag{4-18}$$

(1) 假定 $\sum\limits_{nb} A_{nb}^U u'_{nb} = \sum\limits_{nb} A_{nb}^V v'_{nb} = \sum\limits_{nb} A_{nb}^W w'_{nb} = 0$，则有

$$\begin{cases} u_e = u'_e + u_e^{*} = u_e^{*} + \dfrac{B_e^U}{A_e^U}(p'_E - p'_P), \\ v_n = v'_n + v_n^{*} = v_n^{*} + \dfrac{B_n^V}{A_n^V}(p'_N - p'_P), \\ w_t = w'_t + w_t^{*} = w_t^{*} + \dfrac{B_t^W}{A_t^W}(p'_T - p'_P). \end{cases}$$

计及连续方程

$$\frac{u_e - u_w}{\Delta x} + \frac{v_n - v_s}{\Delta y} + \frac{w_t - w_b}{\Delta z} = 0,$$

得

$$A_P^p p'_P = A_E^p p'_E + A_W^p p'_W + A_N^p p'_N + A_S^p p'_S + A_T^p p'_T + A_B^p p'_B + b, \tag{4-19}$$

其中

$$b = \frac{u_e^{*} - u_w^{*}}{\Delta x} + \frac{v_n^{*} - v_s^{*}}{\Delta y} + \frac{w_t^{*} - w_b^{*}}{\Delta z}, \tag{4-20}$$

$$\begin{cases} A_E^p = \dfrac{B_e^U}{A_e^U \Delta x}, \quad A_W^p = \dfrac{B_w^U}{A_w^U \Delta x}, A_N^p = \dfrac{B_n^V}{A_n^V \Delta y}, \\ A_S^p = \dfrac{B_s^V}{A_s^V \Delta y}, \quad A_T^p = \dfrac{B_t^W}{A_t^W \Delta z}, \quad A_B^p = \dfrac{B_b^W}{A_b^W \Delta z}, \quad A_P^p = \sum_{nb} A_{nb}^p. \end{cases} \tag{4-21}$$

于是在给定 p^* 后可计算 u^*、v^*、w^*；由此计算 b，解方程(4-19)得 p'；以 $p'+p^*$ 代替 p^*，重复计算得 u^*、v^*、w^*；当变化小于 ε 后计算可以停止.

(2) 若假定 $\sum_{nb} A_{nb}^U (u'_{nb} - u'_e) = \sum_{nb} A_{nb}^V (v'_{nb} - v'_n) = \sum_{nb} A_{nb}^W (w'_{nb} - w'_t) = 0$，则有

$$\begin{cases} u_e = u_e^* + \dfrac{B_e^U}{A_e^U - \sum_{nb} A_{nb}^U}(p'_E - p'_P), \\ v_n = v_n^* + \dfrac{B_n^V}{A_n^V - \sum_{nb} A_{nb}^V}(p'_N - p'_P), \\ w = w_t^* + \dfrac{B_t^W}{A_t^W - \sum_{nb} A_{nb}^W}(p'_T - p'_P). \end{cases}$$

这时也可以构成(4-19)式，只是其中的 A_E^p 等有了一些变化，即(4-21)式中分母 A_e^U 改为 $A_e^U - \sum_{nb} A_{nb}^U$ 等.这个方法可以加速收敛，叫作 SIMPLEC 法.

(3) 假定

$$\hat{u}_e = \frac{\sum_{nb} A_{nb}^U U_{nb} + q^U}{A_e^U}, \quad \hat{v}_n = \frac{\sum_{nb} A_{nb}^V V_{nb} + q^V}{A_n^V}, \quad \hat{w}_t = \frac{\sum_{nb} A_{nb}^W W_{nb} + q^W}{A_t^W},$$

并设

$$u_e = \hat{u}_e + u'_e = \frac{\sum_{nb} A_{nb}^U U_{nb} + q^U}{A_e^U} + \frac{B_e^U}{A_e^U - \sum_{nb} A_{nb}^U}(p_E - p_P),$$

$$v_n = \hat{v}_n + v'_n = \frac{\sum_{nb} A_{nb}^V V_{nb} + q^V}{A_n^V} + \frac{B_n^V}{A_n^V - \sum_{nb} A_{nb}^V}(p_N - p_P),$$

$$w_t = \hat{w}_t + w'_t = \frac{\sum_{nb} A_{nb}^W W_{nb} + q^W}{A_t^W} + \frac{B_t^W}{A_t^W - \sum_{nb} A_{nb}^W}(p_T - p_P),$$

类似地得 p 的方程.这种方法称为 SIMPLER 法，更加有效.注意这里直接计算了 p，而不是 p'，用它代替 p^*，再计算下一次的 u^*、v^*、w^*.

应该指出，SIMPLE 法及其改进的方法目前得到广泛的应用.

4.2 非错位网格下 N-S 方程的求解

在任意曲线坐标下,不可压缩流体所满足的方程可以用以下守恒型方程表示出来:

$$J\frac{\partial(\rho\varphi)}{\partial t}+\frac{\partial}{\partial\xi_i}(C_i\varphi-D_{i\varphi})=JS_\varphi,\quad i=1,2,3,\tag{4-22}$$

这里 ξ_i 是曲线坐标系,J 为雅可比行列式,即 $J=\det\left|\frac{\partial(x_1\ x_2\ x_3)}{\partial(\xi_1\ \xi_2\ \xi_3)}\right|$,$x_1,x_2,x_3$ 即 x,y,z,ξ_1,ξ_2,ξ_3 即 ξ,η,ζ.其它项的意义如表 4-1 所示:

表 4-1

No	ϕ	$D_{i\phi}$	C_i	
1	1			0
2	$V_1=u$	$\frac{\Gamma_2}{J}\left(B_j^i\frac{\partial V_1}{\partial\xi_j}+\beta_j^i\omega_1^j\right)$	$\rho\beta_j^iV_j$	$-\frac{1}{J}\frac{\partial}{\partial\xi_j}(\beta_1^jp)$
3	$V_2=v$	$\frac{\Gamma_3}{J}\left(B_j^i\frac{\partial V_2}{\partial\xi_j}+\beta_j^i\omega_2^j\right)$	$\rho\beta_j^iV_j$	$-\frac{1}{J}\frac{\partial}{\partial\xi_j}(\beta_2^jp)$
4	$V_3=w$	$\frac{\Gamma_4}{J}\left(B_j^i\frac{\partial V_3}{\partial\xi_j}+\beta_j^i\omega_3^j\right)$	$\rho\beta_j^iV_j$	$-\frac{1}{J}\frac{\partial}{\partial\xi_j}(\beta_3^jp)$
5	k	$\frac{\Gamma_5}{J}B_j^i\frac{\partial k}{\partial\xi_j}$	$\rho\beta_j^iV_j$	$G-\rho\epsilon$
6	ϵ	$\frac{\Gamma_6}{J}B_j^i\frac{\partial\epsilon}{\partial\xi_j}$	$\rho\beta_j^iV_j$	$(C_{1\epsilon}G-C_{2\epsilon}\rho\epsilon)\frac{\epsilon}{k}$
7	S	$\frac{\Gamma_7}{J}B_j^i\frac{\partial S}{\partial\xi_j}$	$\rho\beta_j^iV_j$	S_S

其中 k 为湍流脉动动能,ϵ 为湍流耗散率,S 为标量(温度,浓度等).

$$\omega_j^i=\beta_j^k\frac{\partial V_i}{\partial\xi_k}\quad\begin{pmatrix}\beta_j^i\text{ 是 }J\text{ 中}\frac{\partial x_i}{\partial\xi_i}\text{ 的}\\\text{代数余子式}\end{pmatrix},J=\begin{vmatrix}\frac{\partial x_1}{\partial\xi_1}&\frac{\partial x_2}{\partial\xi_1}&\frac{\partial x_3}{\partial\xi_1}\\\frac{\partial x_1}{\partial\xi_2}&\frac{\partial x_2}{\partial\xi_2}&\frac{\partial x_3}{\partial\xi_2}\\\frac{\partial x_1}{\partial\xi_3}&\frac{\partial x_2}{\partial\xi_3}&\frac{\partial x_3}{\partial\xi_3}\end{vmatrix},$$

$$B_j^i=\beta_k^i\beta_k^j,\quad\Gamma_n=\mu+\frac{\mu_t}{\sigma_n},\quad\mu_t=\rho\frac{C_\mu k^2}{\epsilon},$$

$$G=\frac{\mu_t}{2J^2}\left(\frac{\partial V_i}{\partial\xi_k}\beta_j^k+\frac{\partial V_j}{\partial\xi_k}\beta_i^k\right)^2,$$

$C_\mu=0.09$, $C_{1\epsilon}=1.44$, $C_{2\epsilon}=1.92$, $\sigma_2=\sigma_3=\sigma_4=1$, $\sigma_5=1$, $\sigma_6=1.3$, $\sigma_7=1$.

网格及其相邻网格中心点如图 4-4.

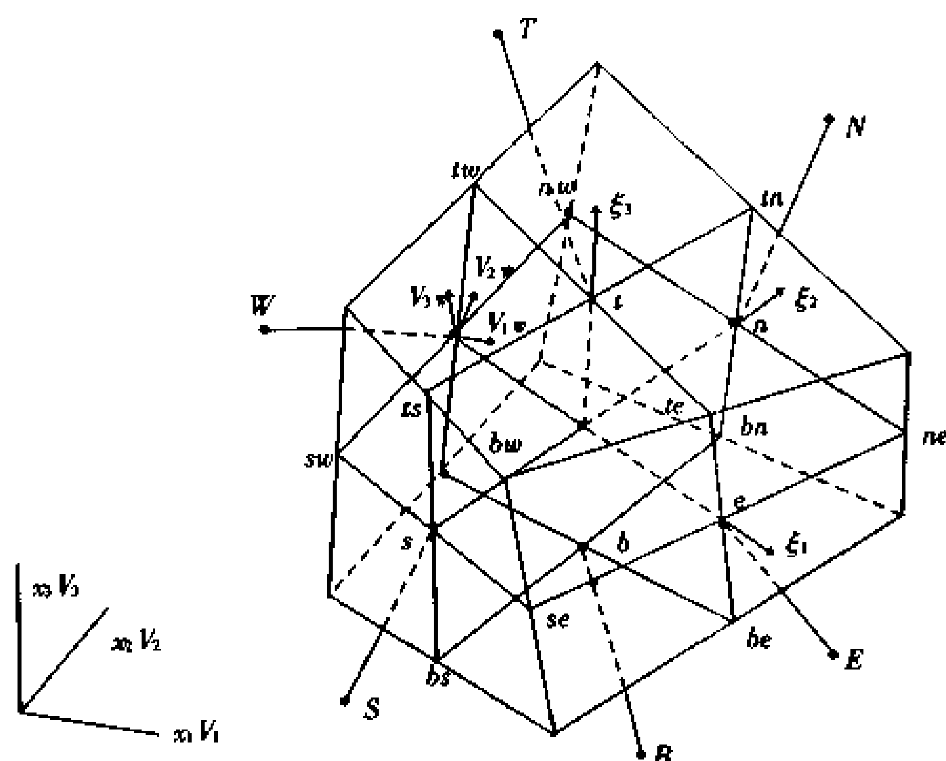


图 4-4

为计算网格交接面中心的值,可采用不同的差分格式.下面以 ξ_1 轴为例说明之(图 4-5).

(1) Hybrid格式: 记 $P_e = |C_w/D_w|$

$$\varphi_w = \begin{cases} 0.5(\varphi_P + \varphi_W), & P_e \leqslant 2, \\ \varphi_W, & P_e > 2. \end{cases} \tag{4-23}$$

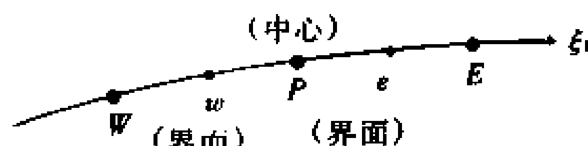


图 4-5

(2) QUICK 格式:

$$\varphi_w = \frac{3}{8}\varphi_P + \frac{3}{4}\varphi_w - \frac{1}{8}\varphi_W. \tag{4-24}$$

(3) SOUCUP 格式:

$$\widehat{\varphi}_w = (\varphi_w - \varphi_W)/(\varphi_p - \varphi_W),$$

$$\varphi_w = \begin{cases} 1.5\varphi_w - 0.5\varphi_W, & 0 < \widehat{\varphi}_w \leqslant 0.5, \\ 0.5(\varphi_P + \varphi_w), & 0.5 \leqslant \widehat{\varphi}_w < 1, \\ \varphi_w, & |\widehat{\varphi}_w - 0.5| \geqslant 0.5. \end{cases} \tag{4-25}$$

(4) HLPA 格式:

$$\varphi_w = \varphi_w + \gamma_w(\varphi_p - \varphi_w)\frac{\varphi_w - \varphi_W}{\varphi_p - \varphi_W}, \tag{4-26}$$

$$\gamma_w = \begin{cases} 1, & |\widehat{\varphi}_w - 0.5| < 0.5, \\ 0, & |\widehat{\varphi}_w - 0.5| \geqslant 0.5. \end{cases}$$

利用方程(4-22)对单元作积分得

$$\int_{\Delta V} \frac{\partial \rho\varphi}{\partial t} \mathrm{d}V + I_e - I_w + I_n - I_s + I_t - I_b = \int_{\Delta V} S_\phi \mathrm{d}V, \tag{4-27}$$

其中 I_f 分成两个部分:对流项部分 I_f^C 和扩散项部分 I_f^D,其中 f 是 e、w、n、s、t、b 之一.它们计算如下:

$$I_f^C = C_f \varphi_f,$$

$$C_w = (\rho b_j^1 V_j)_w, \quad C_s = (\rho b_j^2 V_j)_s, C_b = (\rho b_j^3 V_j)_b,$$

$$\beta_j^i = \frac{b_j^i \Delta \xi_i}{\Delta \xi_1 \Delta \xi_2 \Delta \xi_3},$$

b_j^i,例如

$$b_3^2: b_3^2 = ((x_2)_e - (x_2)_w)((x_1)_t - (x_1)_b) - ((x_1)_e - (x_1)_w)((x_2)_t - (x_2)_b),$$

$$I_f^D = I_f^{DN} + I_f^{DC}.$$

如对于 w 面有

$$\begin{cases} I_w^{DN} = D_w(\varphi_P - \varphi_w), \\ D_w = ((b_1^1 b_1^1 + b_2^1 b_2^1 + b_3^1 b_3^1)\Gamma_\varphi / \Delta V)_w. \end{cases}$$

这是非交错导数项引起的 I^D,I_w^{DC} 为所有交错导数项引起的,应采用上一时间层的值,并移到方程右端作为源项,这里不一一列出.

以上各表达式代入(4-27)式,计及时间方向导数可取

$$\frac{\partial(\rho\varphi)}{\partial t} = \frac{(\rho\varphi)^n - (\rho\varphi)^{n-1}}{\Delta t}$$

或

$$\frac{\partial(\rho\varphi)}{\partial t} = \frac{3(\rho\varphi)^n - 4(\rho\varphi)^{n-1} + (\rho\varphi)^{n-2}}{2\Delta t}.$$

最后得

$$A_P \varphi_P = \sum_{nb} A_{nb} \varphi_{nb} + S_U, \tag{4-28}$$

其中 nb 表示邻居 W、E、N、S、B、T 点,A 为相应的系数,S_U 由 S_φ、I^{DC} 组成.在 k、ϵ 的方程中,计及 k、ϵ 总是正的,S_U 还要作一些小的改动.

在修正速度的过程中,动量插值法即为

$$V_{m,P} = h_P^{1m} + D_m^1(p_w - p_e) + (1 - \alpha_n) V_{m,P}^0, \quad m = 1,2,3, \tag{4-29.a}$$

或

$$V_{m,P} = h_P^{2m} + D_m^2(p_s - p_n) + (1 - \alpha_n) V_{m,P}^0, \quad m = 1,2,3, \tag{4-29.b}$$

或

$$V_{m,P} = h_P^{3m} + D_m^3(p_b - p_t) + (1 - \alpha_n) V_{m,P}^0, \quad m = 1,2,3, \tag{4-29.c}$$

其中 α_n 是一亚松弛系数($0 < \alpha_n \leqslant 1$).

$$\begin{cases} h_p^{1m} = \dfrac{\sum\limits_{nb} A_{nb}^m V_{m,nb} + S^m}{A_p^m} + D_m^2(p_s - p_n) + D_m^3(p_b - p_s) \\ \qquad \xlongequal{\text{记作}} h_p^{(0)} + D_m^2(p_s - p_n) + D_m^3(p_b - p_s), \\ h_p^{2m} = h_p^{(0)} + D_m^1(p_w - p_e) + D_m^3(p_b - p_s), \\ h_p^{3m} = h_p^{(0)} + D_m^1(p_w - p_s) + D_m^2(p_s - p_n), \end{cases} \tag{4-30}$$

而 $V_{m,p}^0$ 是上一次迭代计算得到的速度值，$m = 1,2,3$. 由于压力、速度值不知，所以设为 p^* 和 v^*，于是得

$$\begin{cases} h_p^{1m*} = h_p^{(0)*} + D_m^2(p_s^* - p_n^*) + D_m^3(p_b^* - p_s^*), \\ h_p^{2m*} = h_p^{(0)*} + D_m^1(p_w^* - p_e^*) + D_m^3(p_b^* - p_s^*), \\ h_p^{3m*} = h_p^{(0)*} + D_m^1(p_w^* - p_s^*) + D_m^2(p_s^* - p_n^*), \\ h_p^{(0)*} = \dfrac{\sum\limits_{nb} A_{nb}^m V_{m,nb}^* + S^m}{A_p^m}. \end{cases} \tag{4-31}$$

利用插值的方法，有

$$V_{m,w}^* = V_{m,w}^H + (1 - \alpha_n)(V_{m,w}^0 - V_{m,w}^L),$$

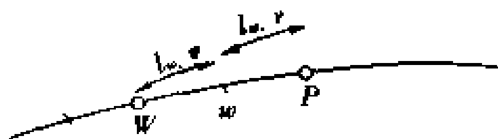
其中

$$V_{m,w}^L = f_\xi^- V_{m,W}^0 + f_\xi V_{m,P}^0 \qquad (V_{m,W}^0\ V_{m,P}^0 \text{ 的插值}),$$

$$V_{m,w}^0 = f_\xi^- V_{m,W} + f_\xi V_{m,P}$$

（$V_{m,W}$ 为新计算得的值）

$$V_{m,w}^H = f_\xi^- H'^{m*}_{P,W} + f_\xi H'^{m*}_P + \left(\frac{f_\xi^-}{A_{P,w}^m} + \frac{f_\xi}{A_P^m}\right)(b'_m)_w(p_W^* - p_P^*)$$



$$f_\xi = \frac{l_{wW}}{l_{w,W} + l_{w,P}}, \quad f_\xi^- = 1 - f_\xi$$

图 4-6

$A_{P,W}$ 是 A_P 在 W 点上的值，$H_{P,W}$ 是 H_P 在 W 处的值，见图4-6. 在 s、b 上有类似的公式. 将(4-29)式和它们中 V,p 改为 V^*,p^* 时的公式相减并假定 $(V_m - V_m^*)_{nb}$ 在 $nb = W,E,S,N,B,T$ 时都与 P 点相等，则得

$$V_{m,P} = V_{m,P}^* + \frac{D'_m(p'_w - p'_e) + D_m^2(p'_s - p'_n) + D_m^3(p'_b - p'_t)}{1 - (\sum\limits_{nb} A_{nb}^m / A_P^m)},$$

其中

$$p'_f = p_f - p_f^* \qquad (f = w,e,s,n,b,t).$$

然后速度修正为

$$V_{m,w}^* = f_\xi^- h_{P,W}^{1m*} + f_\xi h_P^{1m*} + Q_{m,w}^1(p_W^* - p_P^*) + (1 - \alpha_u)V_{m,w}^0,$$

$$V_{m,s}^* = f_\eta^- h_{P,S}^{2m*} + f_\eta h_P^{2m*} + Q_{m,S}^2(p_S^* - p_N^*) + (1 - \alpha_u)V_{m,s}^0,$$

$$V_{m,b}^* = f_\zeta^- h_{P,B}^{3m*} + f_\zeta h_P^{3m*} + Q_{m,b}^3(p_B^* - p_T^*) + (1 - \alpha_u)V_{m,b}^0,$$

$$Q_{m,w}^1=\left(\frac{f_{\xi}^-}{A_{P,W}^m}+\frac{f_{\xi}}{A_P^m}\right)(b_m^1)_W,$$

$$Q_{m,s}^2=\left(\frac{f_{\eta}^-}{A_{P,s}^m}+\frac{f_{\eta}}{A_P^m}\right)(b_m^2)_S,$$

$$Q_{m,b}^3=\left(\frac{f_{\zeta}^-}{A_{P,B}^m}+\frac{f_{\zeta}}{A_P^m}\right)(b_m^3)_B.$$

最后速度值

$$\begin{cases}V_{m,w}=V_{m,w}^*+\dfrac{Q_{m,w}^1}{d_{m,w}^1}(p'_W-p'_P),\\ V_{m,s}=V_{m,s}^*+\dfrac{Q_{m,s}^2}{d_{m,s}^2}(p'_S-p'_N),\\ V_{m,b}=V_{m,b}^*+\dfrac{Q_{m,b}^3}{d_{m,b}^2}(p'_B-p'_T),\end{cases}\tag{4-32}$$

$$\begin{cases}d_{m,w}^1=1-f_{\xi}^-(A^0)_W-f_{\xi}(A^0)_P,\\ d_{m,s}^2=1-f_{\eta}^-(A^0)_S-f_{\eta}(A^0)_P,\\ d_{m,b}^3=1-f_{\zeta}^-(A^0)_B-f_{\zeta}(A^0)_P,\end{cases}\quad A^0=\frac{\sum\limits_{nb}A_{nb}^m}{A_P^m}.\tag{4-33}$$

而所求解的压力方程为

$$C_e-C_w+C_n-C_s+C_b-C_t=0,\tag{4-34}$$

其中

$$\begin{cases}C_w=A_W^p(p'_w-p'_P)+C_w^*, & C_e=A_E^p(p'_P-p'_E)+C_e^*,\\ C_s=A_S^p(p'_S-p'_P)+C_s^*, & C_n=A_N^p(p'_P-p'_N)+C_n^*,\\ C_b=A_B^p(p'_B-p'_P)+C_b^*, & C_t=A_T^p(p'_P-p'_T)+C_t^*,\end{cases}\tag{4-35}$$

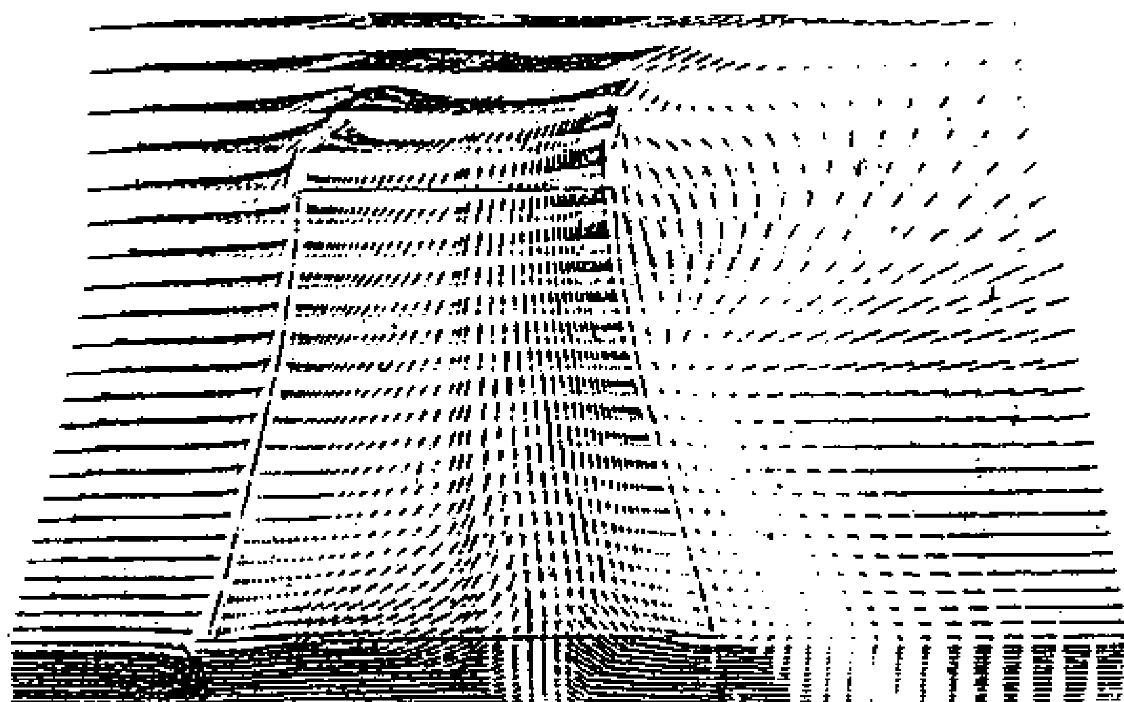
$$\begin{cases}A_W^p=\rho_w(Q_{1w}^1b_{1w}^1+Q_{2w}^1b_{2w}^1+Q_{3w}^1b_{3w}^1),\\ A_S^p=\rho_s(Q_{1s}^2b_{1s}^2+Q_{2s}^2b_{2s}^2+Q_{3s}^2b_{3s}^2),\\ A_B^p=\rho_b(Q_{1b}^3b_{1b}^3+Q_{2b}^3b_{2b}^3+Q_{3b}^3b_{3b}^3),\end{cases}\tag{4-36}$$

$$\begin{cases}C_w^*=C_w^H+(1-\alpha_u)(C_w^0-C_w^L),C_w^H=\rho_w(b_{1w}^1V_{1w}^H+b_{2w}^1V_{2w}^H+b_{3w}^1V_{3w}^H),\\ C_s^*=C_s^H+(1-\alpha_u)(C_s^0-C_s^L),C_s^H=\rho_s(b_{1s}^2V_{1s}^H+b_{2s}^2V_{2s}^H+b_{3s}^2V_{3s}^H),\\ C_b^*=C_b^H+(1-\alpha_u)(C_b^0-C_b^L),C_b^H=\rho_s(b_{1b}^3V_{1b}^H+b_{2b}^3V_{2b}^H+b_{3s}^3V_{3s}^H),\\ C_w^LC_s^LC_b^L\text{ 为 }C_w^HC_s^HC_b^H\text{ 中的 }V^H\text{ 相应地改为 }V^L.\end{cases}\tag{4-37}$$

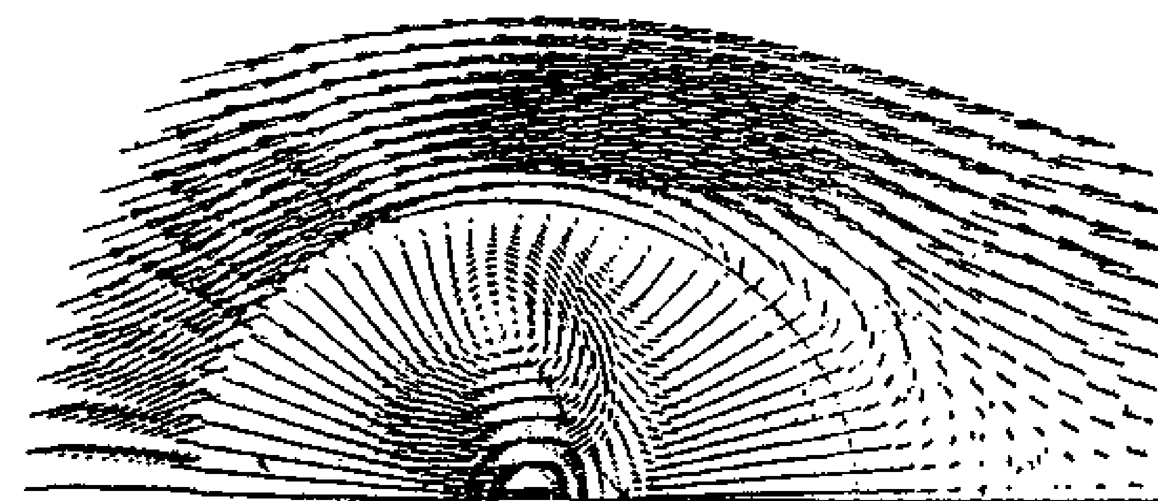
将(4-34)式详细写出后得到方程

$$\begin{cases}A_P^pp'_P=A_E^pp'_E+A_W^pp'_W+A_N^pp'_N+A_S^pp'_S+A_B^pp'_B+A_T^pp'_T-\Delta Q,\\ \Delta Q=C_e^*-C_w^*+C_n^*-C_s^*+C_b^*-C_t^*.\end{cases}\tag{4-38}$$

该方程可以用迭代法或强隐格式求解.不难看出这就是前面讨论过的 SIMPLEC 的作法.该方法得到的程序已经发展为一个大型软件包.用该软件包计算空冷塔在侧向风作用下的流场和温度场如图 4-7 所示.



(a) 空冷塔在侧向风作用下子午面内的速度场



(b) 在水平面内的速度场

(c) 在水平面内$(T-T_a)/(T_{w1}-T_a)$的分布

图 4-7

4.3 BGK 法解不可压缩粘性流体流场

在求解不可压缩流体流动时，有一种常用的方法是伪压缩法，即将流动看作是可压缩的，只是压缩性很差，这时方程写作

$$
\begin{cases}\dfrac{1}{\beta}\dfrac{\partial \rho}{\partial t}+\nabla\cdot V=0,\\[2mm] \dfrac{\partial V}{\partial t}+(V\cdot\nabla)V=-\dfrac{1}{\rho}\nabla p+\nu\Delta V.\end{cases}\tag{4-39}
$$

当 $\beta\to\infty$ 时就是原来不可压缩流体流动的N-S方程.但无论 β 有多大，总是带来一定的误差，而且 β 越大，计算越困难.为此人们用马赫数 M 比较小的流动来近似不可压缩的流动，一般认为马赫数小于0.15时就可以近似地看作不可压缩流动.事实上如果利用前面讨论过的BGK方法，不难将它推广到不可压缩的流动中去.在不可压缩流动中没有强间断，所以可压缩流体计算中的BGK法，计算中交接面的左、右侧是相同的，在交接面上宏观量的导数的计算为

$$
s_{j+\frac{1}{2}}=\frac{7}{12}(s_j+s_{j+1})-\frac{1}{12}(s_{j-1}+s_{j+2}),
$$

$$
\left(\frac{\mathrm{d}s}{\mathrm{d}x}\right)_{j+\frac{1}{2}}=\frac{\dfrac{5}{4}(s_{j+1}-s_{j-1})-\dfrac{1}{12}(s_{j+2}-s_{j-1})}{\Delta x}.
$$

前面讨论的计算通量的公式都可以用，而且更加简化了，如(3-171)、(3-172)式已不再需要，$a_r=a_l$，$=\bar{a}_r=\bar{a}_l=a$ 且它的三个分量 a_1、a_2、a_3 为

$$
a_3=\frac{4\lambda^2}{K+1}(\tilde{B}-2U\tilde{A}),\quad a_2=2\lambda\left(\tilde{A}-\frac{a_3U}{2\lambda}\right),
$$

$$
a_1=\frac{1}{\rho}\frac{\partial\rho}{\partial x}-a_2U-a_3\left(\frac{U^2}{2}+\frac{K+1}{4\lambda}\right).
$$

其中 $\tilde{A}=\dfrac{1}{\rho}\left[\dfrac{\partial(\rho U)}{\partial x}-U\dfrac{\partial\rho}{\partial x}\right]$，$\tilde{B}=\dfrac{1}{\rho}\left[\dfrac{\partial(\rho e)}{\partial x}-\left(U^2+\dfrac{K+1}{2\lambda}\right)\dfrac{\partial\rho}{\partial x}\right]$.

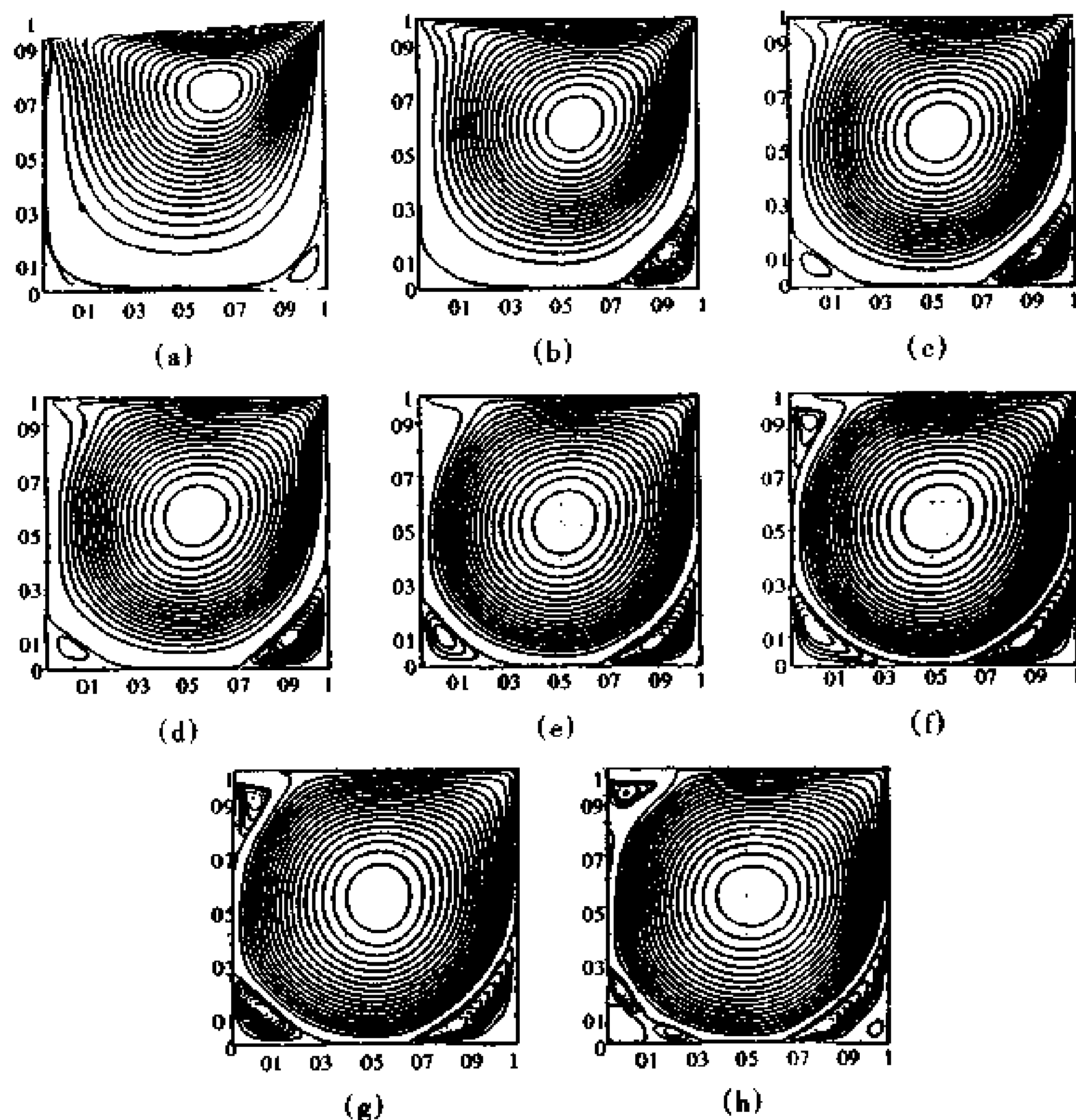
而 A 可以如下确定：

$$
MA=\begin{bmatrix}1 & U & \frac{1}{2}\left(U^2+\frac{K+1}{2\lambda}\right)\\ U & U^2+\frac{1}{2\lambda} & \frac{1}{2}\left(U^3+\frac{(K+3)U}{2\lambda}\right)\\ \frac{1}{2}\left(U^2+\frac{K+1}{2\lambda}\right) & \frac{1}{2}\left(U^3+\frac{(K+3)U}{2\lambda}\right) & \frac{1}{2}\left(U^4+\frac{(K+3)U^2}{\lambda}\right)+\frac{K^2+4K+3}{4\lambda^2}\end{bmatrix}\begin{bmatrix}A_1\\ A_2\\ A_3\end{bmatrix}
$$

$$
=\frac{1}{\rho}\int u(a_1,a_2,a_3)\begin{bmatrix}1\\ u\\ \frac{1}{2}(u^2+\xi^2),\end{bmatrix}g_0\mathrm{d}u\mathrm{d}\zeta.
$$

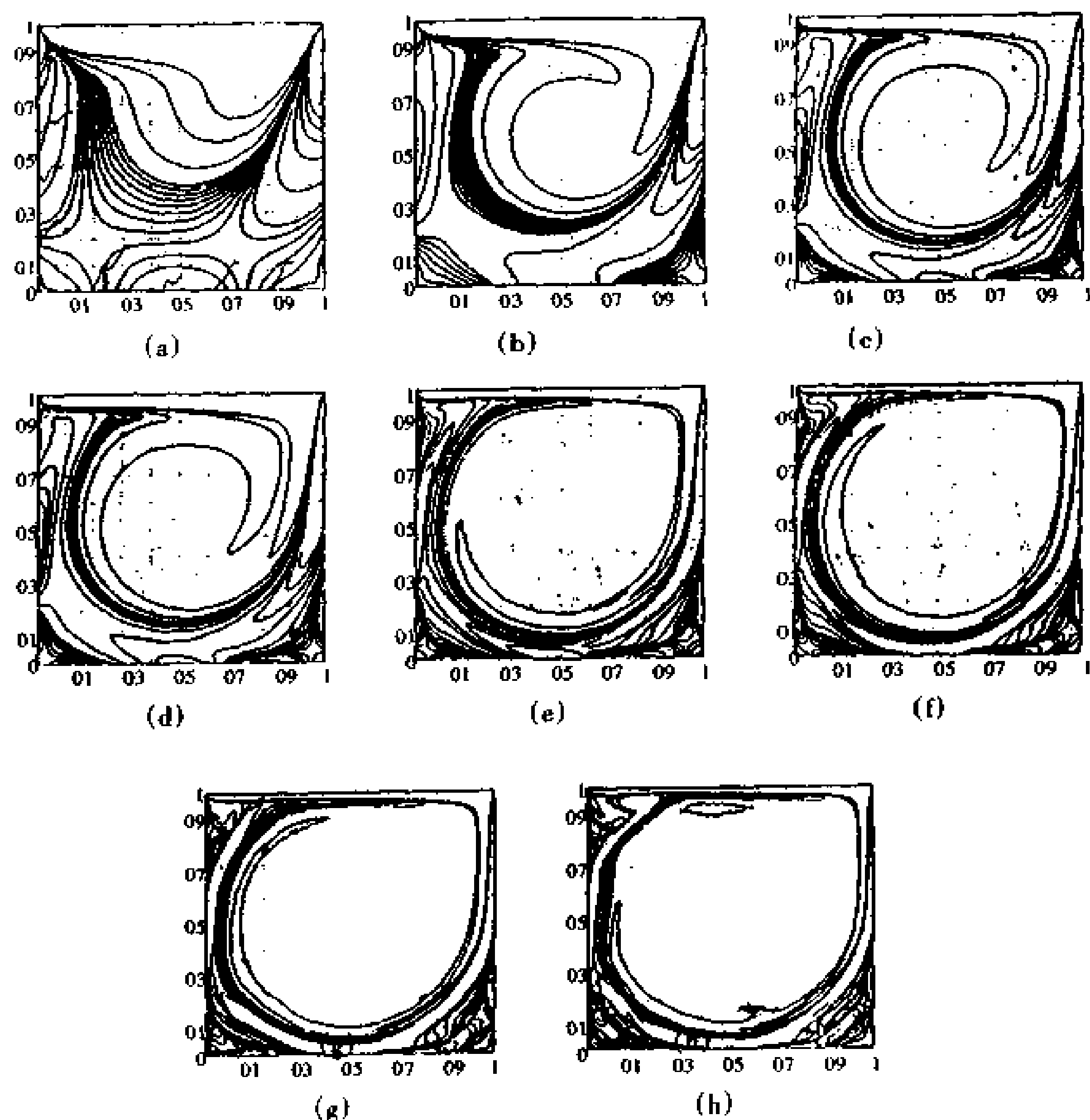
以上是对一维而言的.对于二维、三维情况也一样.在计算不可压流场时，经过计算方框流，并与Ghia的结果比较表明，$K=0$，$M\leqslant 0.15$ 时即可得到与不可压缩流场

用 ψ-ω 方法得到的结果相同.所以 BGK 方法在这样的参数下计算不可压缩流场是完全可以的.图 4-8、图 4-9、图 4-10 给出了方框流的流线、涡线及速度剖面结果.



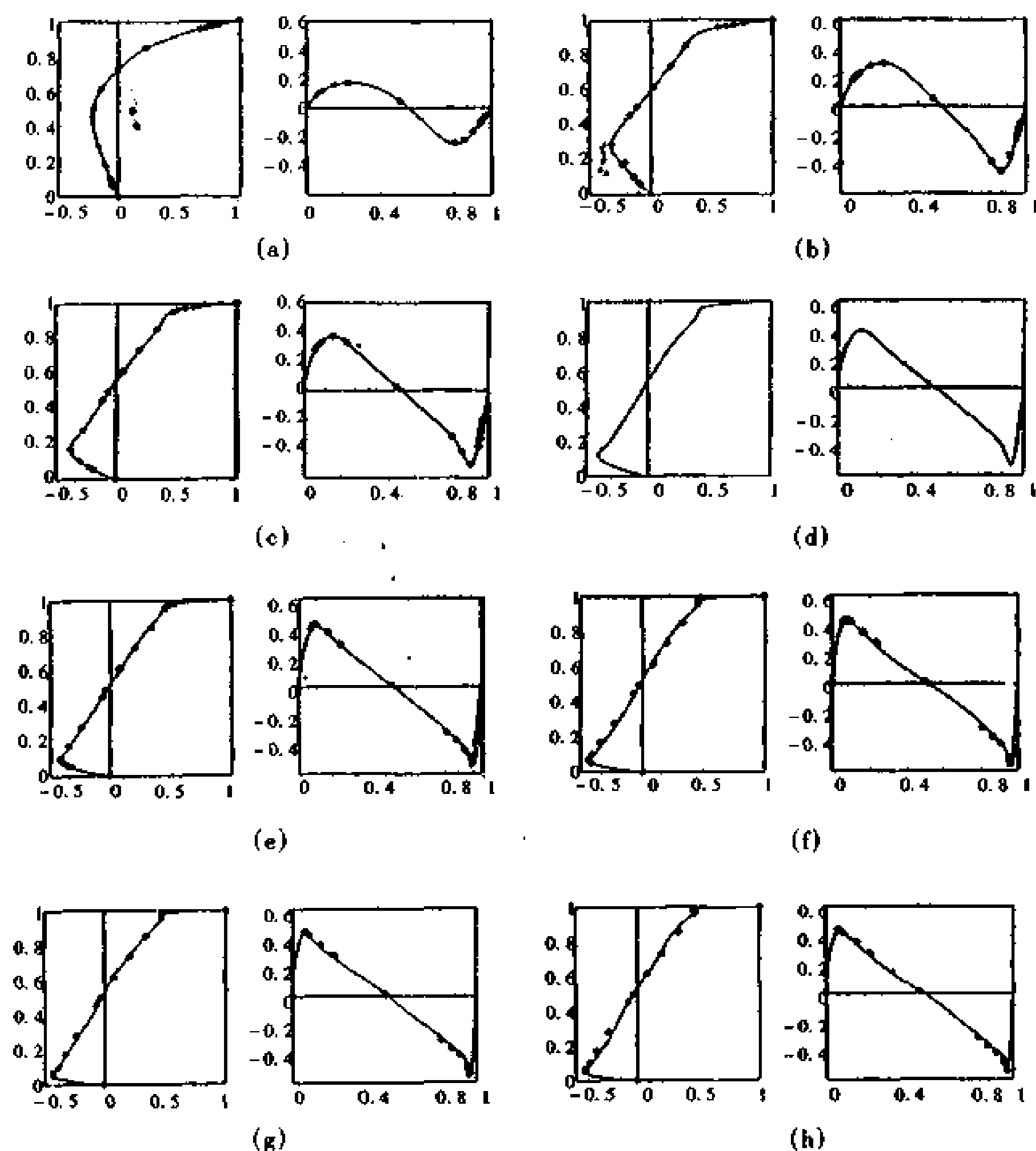
(a) Re = 100; (b) Re = 400; (c) Re = 1000; (d) Re = 2000;
(e) Re = 3200; (f) Re = 5000; (g) Re = 7500; (h) Re = 10000

图 4-8



(a) Re = 100;　(b) Re = 400;　(c) Re = 1000;　(d) Re = 2000;
(e) Re = 3200;　(f) Re = 5000;　(g) Re = 7500;　(h) Re = 10000

图 4-9



(a) Re = 100;　(b) Re = 400;　(c) Re = 1000;　(d) Re = 2000;
(e) Re = 3200;　(f) Re = 5000;　(g) Re = 7500;　(h) Re = 10000

图 4-10

图4-11是用 *BGK* 格式计算的自由剪切流的流场,它真实地反映了涡的结构和发展.

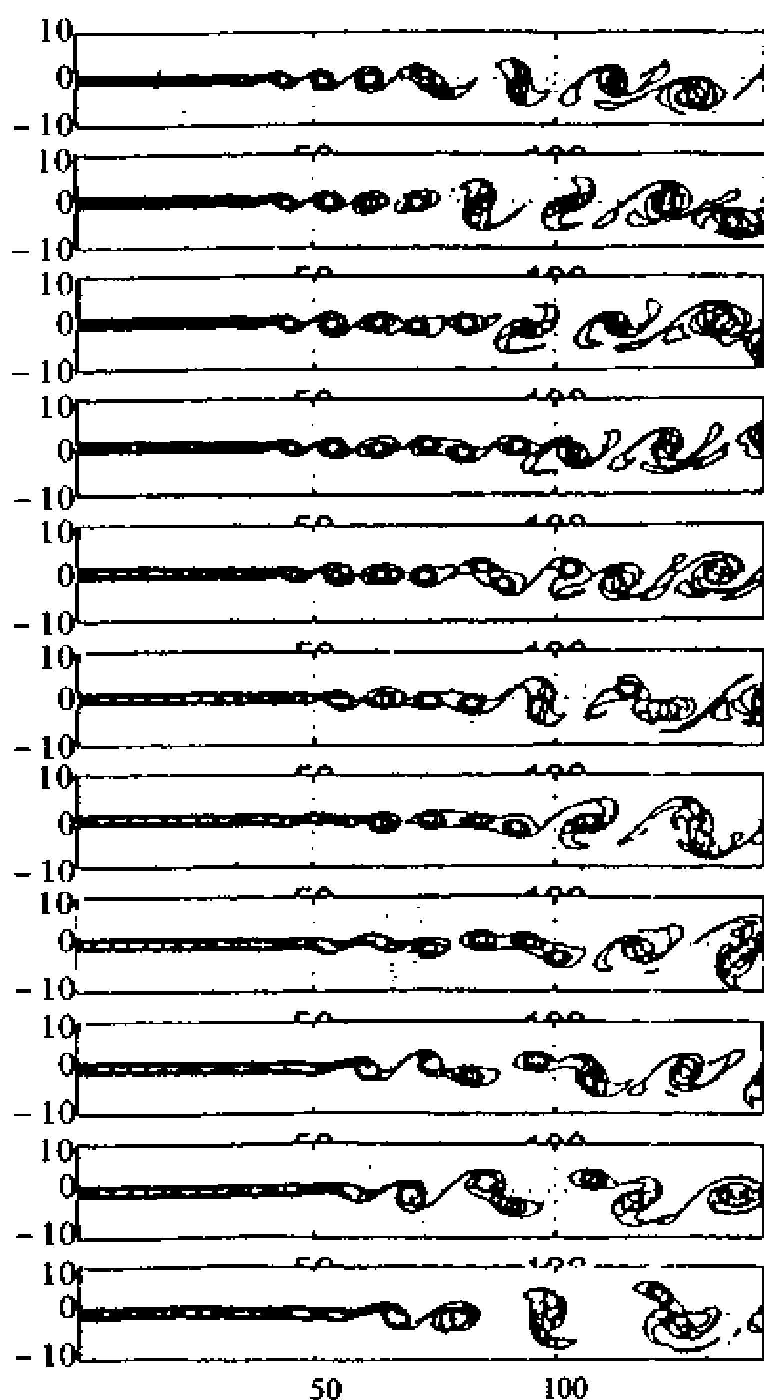


图 4-11

5 高精度格式

大家知道，为了数值模拟湍流运动需要大的计算机，即超级巨型机，并且许多台这样的超级巨型机并行计算才行. 即使如此，为了分辨湍流运动中的最小尺度的

涡,要求网格的尺度必须小于该尺度.湍流理论的分析表明,对于雷诺数为 Re 的流动,相应的湍流中的最小的涡的尺度与科尔莫戈罗夫(Kolmogorov)涡的尺度相当,这就是说网格的尺度应当是 $L/Re^{3/4}$;其中 L 为确定雷诺数的特征尺度.如果设计算域的尺度与特征尺度可比拟,则网格数在一个坐标方向上就应当有 $Re^{3/4}$ 个,在三个方向上总的网格数就应当有 $Re^{9/4}$ 个,对于 $Re = 10000$ 的情况,网格数至少应当有 $1000*1000*1000 = 10^9$ 个(十亿!).现在的世界上最大最快的计算机也只能用来模拟 $Re \sim 10000$ 的湍流运动.而事实上大自然和工业上遇到的湍流运动最大量级是 10^7 以上,在大气运动中则约为 10^9,所以为了满足湍流数值模拟的需要,人们至少还要等上 50 年.

所以人们为了在有限容量和速度的计算机条件下对尽可能高雷诺数的湍流运动进行数值模拟,除了大力加速计算机的发展以外,还需要不断改进数值计算方法,力求不断提高数值计算的精度、分辨率和计算的效率,使有限的计算机资源发挥最大的效解.本章着重介绍紧致高精度格式.

分析一个差分格式的精度,最常用的方法是采用泰勒展开的方法.如格式

$$\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_i = \sum_{k=-K_1}^{K_2} \alpha_{i+k} f_{i+k} + O(\Delta x^p) \qquad (K_1, K_2 \geqslant 0) \tag{5-1}$$

的精度是将 $f(x)$ 函数在 $x = x_i$ 处进行泰勒展开,而 $f(x_{i+k}) = f(x_i + k\Delta x)$,故将该展开式中的 $x - x_i$ 用 $k\Delta x$ 代替即得 f_{i+k} 的表达式.将所有这些表达式代入(5-1)式,比较(5-1)式左右两边,其中差别为 $O(\Delta x^p)$,则称该格式为 p **阶精度的**.通常的格式如下:

$$\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_i = \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x) \tag{5-2}$$

是一阶精度的,称作一阶格式.该格式涉及 i、$i+1$ 两个点叫做**下风格式**.

$$\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2) \tag{5-3}$$

是二阶精度的,涉及前后两个点叫做**中心差分格式**.一般的差分格式可以这样来构造:设

$$f(x) \approx \sum_{j=1}^{N} f_j \frac{\prod\limits_{i \neq j}^{N} (x - x_i)}{\prod\limits_{i \neq j}^{N} (x_j - x_i)}, \tag{5-4}$$

这是一个拉格朗日多项式,对 $f(x)$ 求一阶导数,并令 $x = x_j$,即可得到 $(\mathrm{d}f/\mathrm{d}x)_j$ 的差分格式.求二阶导数并令 $x = x_j$,可得 $(\mathrm{d}^2 f/\mathrm{d}x^2)_j$ 的差分格式.格式的精度与求导后得到的多项式次数有关,还和点的位置有关.采用这样方法得到的各阶精度的差分格式可见本篇篇末所附"平衡态分布函数矩(A)".不难看到,随着精度的提高涉及的点就越多,这对于边界点及接近边界的点的导数计算带来很大的困难,所以人们转向采用比较少的点而又使解达到同样精度的紧致格式.比如要计算一阶导数,可以采用如下的格式:

$$\sum_{k=-K_1^{(1)}}^{K_2^{(1)}} \alpha_k F_{j+k} = \sum_{k=-K_1^{(0)}}^{K_2^{(0)}} a_k f_{j+k}, \tag{5-5}$$

其中 $F_{j+k} = \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_{j+k}$，$j+k$ 表示在 $x = x_{j+k}$ 处的值.对于网格等间距的情况，有 $x_{j+k} = x_j + k\Delta x$.为确定 α_k(这里共有 $2 + K_1^{(1)} + K_2^{(1)} + K_1^{(0)} + K_2^{(0)}$)，采用泰勒展开的方法，即

$$\begin{aligned} F_{j+k} &= \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_{j+k} = \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_j + \left(\frac{\mathrm{d}^2f}{\mathrm{d}x^2}\right)_j k\Delta x + \cdots + \left(\frac{\mathrm{d}^{p+1}f}{\mathrm{d}x^{p+1}}\right)_j \frac{k^p\Delta x^p}{p!} + \cdots, \\ f_{j+k} &= f_j + \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_j k\Delta x + \cdots + \left(\frac{\mathrm{d}^pf}{\mathrm{d}x^p}\right)_j \frac{k^p\Delta x^p}{p!} + \cdots, \end{aligned} \tag{5-6}$$

将它代入(5-5)式可以得到

$$\sum_{k=-K_1^{(1)}}^{K_2^{(1)}} \alpha_k \sum_{l=1}^{\infty} \frac{1}{(l-1)!} k^{l-1}\Delta x^{l-1}\left(\frac{\mathrm{d}^lf}{\mathrm{d}x^l}\right)_j = \sum_{k=-K_1^{(0)}}^{K_2^{(0)}} a_k \sum_{l=0}^{\infty} \frac{1}{l!}\left(\frac{\mathrm{d}^lf}{\mathrm{d}x^l}\right)_j,$$

或改写为

$$\sum_{l=1}^{\infty}\left(\sum_{k=-K_1^{(1)}}^{K_2^{(1)}} \alpha_k \frac{k^{l-1}\Delta x^{l-1}}{(l-1)!}\left(\frac{\mathrm{d}^lf}{\mathrm{d}x^l}\right)_j\right) = \sum_{l=0}^{\infty}\left(\sum_{k=-K_1^{(0)}}^{K_2^{(0)}} a_k \frac{k^l\Delta x^l}{l!}\left(\frac{\mathrm{d}^lf}{\mathrm{d}x^l}\right)_j\right).$$

比较导数得

$$\sum_{k=-K_1^{(0)}}^{K_2^{(0)}} a_k = 0,$$

$$\sum_{k=-K_1^{(1)}}^{K_2^{(1)}} \alpha_k \frac{k^{l-1}\Delta x^{l-1}}{(l-1)!} = \sum_{k=-K_1^{(0)}}^{K_2^{(0)}} a_k \frac{k^l\Delta x^l}{l!}, \quad l = 1,2,\cdots. \tag{5-7}$$

由于得到的关于 α_k、a_k 方程都是各次的线性方程组，所以为确定起见，l 应当小于等于 $K_1^{(1)} + K_2^{(1)} + K_1^{(0)} + K_2^{(0)}$，而再附加一个其它的条件，如 $\alpha_1 = 1$ 等.对于生成二阶导数的格式的方法与上是相同的，只是所求的是二阶导数而已.在(5-7)式中若最大的 l 等于 p，则精度就是 $(p-1)$ 阶的.对于二阶导数格式而言也是一样.精度是 $(p-2)$ 阶的.利用以上方法可以构造各式各样的高精度格式.由于采用点少，所以叫做紧致高精度格式.附录中给出了一些常用的紧致的高精度格式.关于它们的精度可以通过泰勒展开并比较两端相应的导数，由余量的阶数得到精度的阶数.读者可以作为练习自行分析.

导出紧致格式的另一种方法是采用帕德(Padé)逼近的方法.大家知道，帕德逼近有精度高、收敛快等特点，在数值分析中有广泛的应用.它的思想可以应用于微分方程求解中来.帕德逼近内容很丰富，希望了解这方面的工作可以参考有关参考书和文献，这里仅作十分简单的介绍.

设 $f(x) = \ln(1+x)$，它的泰勒级数为

$$f(x)=\sqrt{\frac{1+2x}{1+x}}=1+\frac{1}{2}x-\frac{5}{8}x^2+\frac{13}{16}x^3-\frac{141}{128}x^4+\cdots.\qquad(5\text{-}8)$$

当 x 接近1/2时，上面级数收敛非常慢；当 $|x|\geqslant\frac{1}{2}$ 时，级数就不收敛了. 为了改善对函数的逼近，可以设

$$\ln(1+x)=\frac{P(x)}{Q(x)}+O(x^{\alpha}),\qquad(5\text{-}9)$$

其中 $P(x),Q(x)$ 分别是 m、n 次多项式. $O(x^{\alpha})$ 是 α 次及更高次的项的总和. 设

$$P(x)=\sum_{i=0}^{m}a_ix^i,\qquad Q(x)=\sum_{i=0}^{n}b_ix^i,\qquad(5\text{-}10)$$

另改写(5-8)式为

$$\ln(1+x)=T_{\alpha}(x)+\epsilon_T(x),\qquad(5\text{-}11)$$

其中 $T_{\alpha}(x)$ 为(5-8)式中直到 $x^{\alpha-1}$ 次的所有项之和(即前 α 项之和)，$\epsilon_T(x)$ 为其余项，或计作

$$T_{\alpha}(x)=\sum_{i=0}^{\alpha}C_ix^i.\qquad(5\text{-}12)$$

将(5-11)式代入(5-9)式，两边乘以 $Q(x)$ 得

$$T_{\alpha}(x)Q(x)-P(x)=O(x^{\alpha+n}),\qquad(5\text{-}13)$$

其中 $\epsilon_T(x)$ 等已吸收到 $O(x^{\alpha+n})$ 中去了. 为了方便起见，设 $b_0=1$，并选 $\alpha=m+1$，则上式比较导数可以得到

$$\begin{bmatrix}a_0\\a_1\\a_2\\\vdots\\a_m\end{bmatrix}=\begin{bmatrix}c_0&0&0&\cdots&0\\c_1&c_0&0&\cdots&0\\c_2&c_1&c_0&\cdots&0\\\vdots&\vdots&\vdots&&\vdots\\c_m&c_{m-1}&c_{m-2}&\cdots&c_{m-n}\end{bmatrix}\begin{bmatrix}b_0\\b_1\\b_2\\\vdots\\b_n\end{bmatrix},\qquad(5\text{-}14)$$

$$\begin{bmatrix}c_{m+1}&c_m&\cdots&c_{m-n+1}\\c_{m+2}&c_{m+1}&\cdots&c_{m-n+2}\\\vdots&\vdots&&\vdots\\c_{m+n}&c_{m+n-1}&\cdots&c_m\end{bmatrix}\begin{bmatrix}b_0\\b_1\\\vdots\\b_n\end{bmatrix}=\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix},\qquad(5\text{-}15)$$

这里设 $m\geq n$. 如设 $m=n$，则在不同的 $m=n$ 值可以得到

$$m=n=0,\quad P(x)/Q(x)=1,$$

$$m=n=1,\quad \frac{P(x)}{Q(x)}=\frac{1+\frac{7}{4}x}{1+\frac{5}{4}x},$$

$$m=n=2,\quad \frac{P(x)}{Q(x)}=\frac{1+\frac{13}{4}x+\frac{41}{16}x^2}{1+\frac{11}{4}x+\frac{29}{16}x^2}.$$

可以发现，当 $x \to \infty$ 时，上述逼近式的值分别为

$$1,\ 1.4,\ 1.413793103,\ 1.414201183$$

很快收敛到$\sqrt{2}$. 可见上面的方法大大加速了收敛性，还扩大了收敛域. 这个想法可以应用到求导数上，比如

$$F = \mathrm{D}f, \tag{5-16}$$

其中 $\mathrm{D}f$ 即表示$\frac{\mathrm{d}f}{\mathrm{d}x}$，这一表示式也可以改写为

$$F = \frac{P(\mathrm{D})}{Q(\mathrm{D})}f, \tag{5-17}$$

其中

$$\begin{cases} Q(\mathrm{D}) = 1 + b_1\mathrm{D} + b_2\mathrm{D}^2 + \cdots + b_n\mathrm{D}^n, \\ P(\mathrm{D}) = a_0 + a_1\mathrm{D} + \cdots + a_m\mathrm{D}^m. \end{cases} \tag{5-18}$$

利用牛顿公式可得

$$\begin{aligned} \frac{P(\mathrm{D})}{Q(\mathrm{D})} &= (a_0 + a_1\mathrm{D} + \cdots + a_m\mathrm{D}^m) \cdot [1 - (b_1\mathrm{D} + b_2\mathrm{D}^2 + \cdots + b_n\mathrm{D}^n) + \\ &\quad (b_1\mathrm{D} + b_2\mathrm{D}^2 + \cdots + b_n\mathrm{D}^n)^2 + \cdots \\ &= c_0 + c_1\mathrm{D} + c_2\mathrm{D}^2 + \cdots + c_\alpha\mathrm{D}^\alpha + \cdots, \end{aligned}$$

比较导数，并使 $c_0 = 0, c_1 = 1, c_2 = \cdots = c_\alpha = 0$，则这一个逼近就是 $\alpha + 1$ 阶的逼近. 所逼近的的值 F 为一阶导数，如 $c_0 = c_1 = 0, c_2 = 1, c_3 = \cdots = c_\alpha = 0$，则所逼近的就是二阶导数了. 利用这样的关系可以得到

$$Q(\mathrm{D})F = P(\mathrm{D})f. \tag{5-19}$$

将 D用$\delta^0 = (\delta^+ + \delta^-)/2$代替. D^2用$\delta^2 = \delta^+ \cdot \delta^-$代替等等，则可以得到一个紧致的格式. 如

$$F = \frac{\mathrm{D} - 2\varepsilon\mathrm{D}^2}{1 + 2\varepsilon\mathrm{D} + \frac{1}{6}\mathrm{D}^2}f \tag{5-20}$$

是三阶的，所得的格式为

$$\left(1 + 2\varepsilon\mathrm{D} + \frac{1}{6}\mathrm{D}^2\right)F_j = (\mathrm{D} - 2\varepsilon\mathrm{D}^2)f_j \tag{5-21}$$

在 $\epsilon = 1/6$ 时就是本篇末所附“平衡态分布函数矩”(B) 中三阶下风格式了. 其中 $\delta^+ f_j = (f_{j+1} - f_j)/\Delta$; $\delta^- f_j = (f_j - f_{j-1})/\Delta$. 有关进一步的细节可以参考 Collatz 的“The numerical freatment of differential equations”.

差分格式对不同频率的分量产生的误差是不一样的，它提供了一个分析各种差分近似的分辨率特性的有效工具. 这里所谓的分辨率是指时间方向的. 而且通过分析可以帮助我们找到最优的差分格式，还可以对多维情况下不同方向不同特性的各向异性性进行分析.

下面讨论 $f(x)$ 的一个波数 k 的分量，则有

$$f(x) \sim \mathrm{e}^{\frac{2\pi kx}{L}i} \tag{5-22}$$

它的导数为

$$f'(x) \sim \frac{2\pi k}{L}\mathrm{i}\exp\left(\frac{2\pi kx}{L}\mathrm{i}\right),\quad f''(x) \sim -\left(\frac{2\pi k}{L}\right)^2\exp\left(\frac{2\pi kx}{L}\mathrm{i}\right),$$

代入中心差分格式

$$F_j = \frac{f_{j+1} - f_{j-1}}{2\Delta x},$$

右端项为

$$\frac{i}{\Delta x}\sin\frac{2\pi k}{L}\Delta x\exp\left(\frac{2\pi kx}{L}\mathrm{i}\right), \tag{5-23}$$

而左边则为

$$\frac{2\pi k}{L}\mathrm{i}\exp\left(\frac{2\pi kx}{L}i\right) = \frac{2\pi k}{N\Delta x}\exp\left(\frac{2\pi kxj}{L}\mathrm{i}\right). \tag{5-24}$$

可以看出，采用谱方法时，可以得到准确的导数. 而用二阶中心差分时得到的是(5-23)式. 在相同的 L、Δx 的情况下，不同的 k 有不同的精度. 当 $k\to 0$ 时(5-23)式与(5-24)式是相近的. 但当 k 增加时它们的差别就变大. 在图 5-1 一阶导数不同格式波数与精度的关系中可以看到它们的差别，其中，不失一般性，设 $\frac{2\pi}{L}=1$. 在本篇末所附"平衡态分布函数矩(B)"中中心格式(1)的系数应为

$$\frac{a\sin\frac{2\pi\Delta x}{L}R + \frac{b}{2}\sin\left(\frac{4\pi\Delta x}{L}R\right) + \frac{c}{3}\sin\left(\frac{6\pi\Delta x}{L}R\right)}{1 + 2\alpha\cos\frac{2\pi\Delta x}{L}R + 2\beta\cos\left(\frac{4\pi\Delta xk}{L}\right)}, \tag{5-25}$$

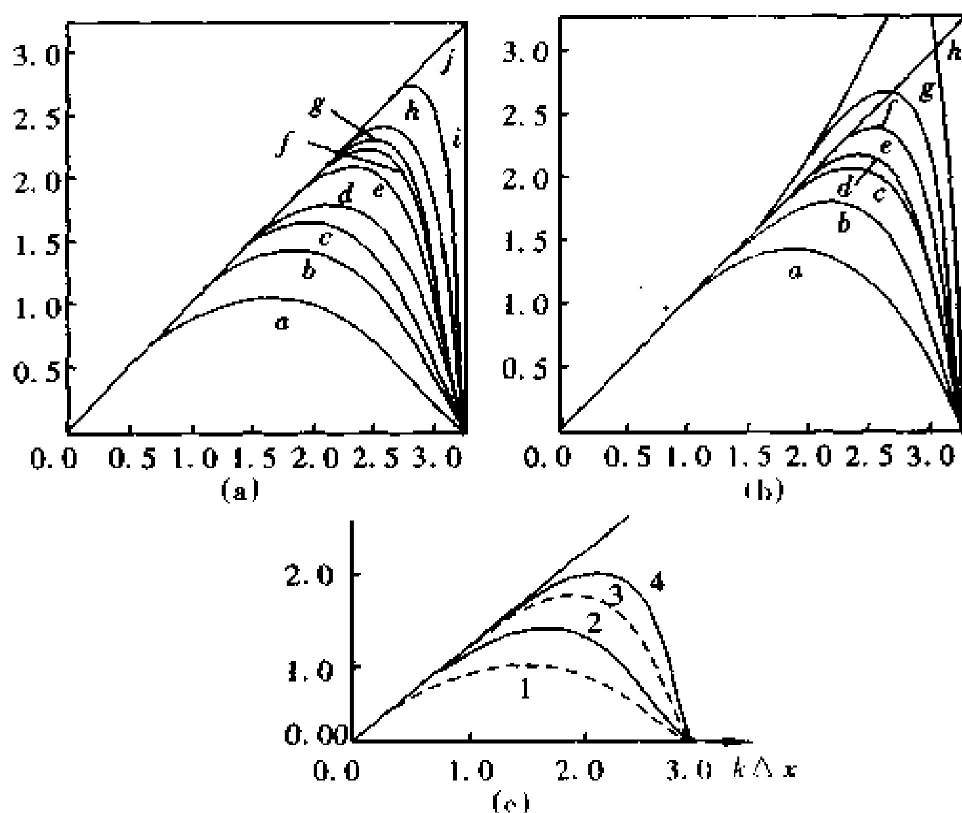


图 5-1

即图 5-1 中的 b，当格式不同时相应的系数也不同. 在图 5-1 中可以看到当精度

提高时,与精度解逼近的 k 范围也就大,这表明对比较高的波数的分量,计算误差比较大,因为波数大的波的周期比较短,这种短波长的波,只有用谱方法才能完全识别,而一般精度的格式的分辨率比较低,这里的高精度格式具有很高的分辨率.也称有谱一样的高分辨率,(5-25)式是由三个参数的四阶精度的格式得到的.在适当选三个参数时,它的分辨率可以和谱方法接近.另外需要注意的是并非精度越高,一定分辨率也高,在图5-1中也说明了这点.

同样的分析也可以对二阶导数进行,这时的三参数格式的系数为

$$\frac{2a\left(1-\cos\frac{2\pi k\Delta x}{L}\right)+\frac{b}{2}\left(1-\cos\frac{4\pi k\Delta x}{L}\right)+\frac{2c}{9}\left(1-\cos\frac{6\pi k\Delta x}{L}\right)}{1+2\alpha\cos\frac{2\pi k\Delta x}{L}+\beta\cos\frac{4\pi k\Delta x}{L}}, \quad (5\text{-}26)$$

在精确的情况下,系数与 k^2 成正比,图5-2(二阶导数不同格式波数与精度关系)看到分辨率的情况.

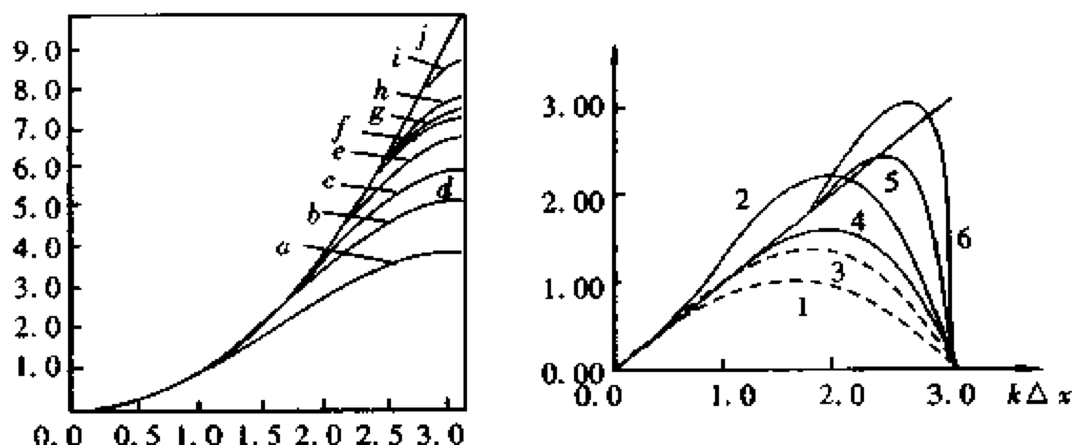


图5-2

要求最佳参数 α、β, a 是比较困难的,但可以提供下列数据作为参考:

一阶导数 $\alpha = 0.5771439, \beta = 0.0896406, a = 1.3025116$

$b = 0.9935500, \quad c = 0.03750245$

二阶导数 $\alpha = 0.5771439, \beta = 0.05569169, a = 10.21564935$

$b = 1.7233220, \quad c = 0.17659730$

关于边界条件在高阶精度格式计算时,边界上的导数计算是比较困难的,一般采用降价的方法,即降低计算精度的要求.这里给出一种二点式的格式,即

$$F_1+\alpha F_2=\frac{1}{\Delta x}(af_1+bf_2+cf_3+df_4). \quad (5\text{-}27)$$

为保证三阶精度,则

$$a=-\frac{11+2\alpha}{6}, \quad b=\frac{6-\alpha}{2}, \quad c=\frac{2-\alpha}{2}, \quad d=\frac{2-\alpha}{6}; \quad (5\text{-}28)$$

四阶精度,则

$$a=3,a=-\frac{17}{6},b=\frac{3}{2},c=\frac{3}{2},d=-\frac{1}{6};\tag{5-29}$$

二阶精度，则

$$a=-\frac{1}{2}(3-\alpha+2d),b=2+3d,c=-\frac{1}{2}(1-\alpha+6d).\tag{5-30}$$

二阶导数的边界条件则是

$$S_1+11S_2=\frac{1}{\Delta x^2}(13f_1-27f_2+15f_3-f_4),\tag{5-31}$$

它是三阶精度的.另外还可以有

二阶精度

$$S_1+11S_2=\frac{1}{\Delta x^2}(13f_1-27f_2+15f_3-f_4),\tag{5-32}$$

三阶精度

$$S_1=\frac{1}{\Delta x^2}\left(\frac{35}{12}f_1+\frac{26}{3}f_2+\frac{19}{2}f_3-\frac{14}{3}f_4-\frac{11}{12}f_5\right).\tag{5-33}$$

也可以设

$$S_1+\alpha S_2=\frac{1}{\Delta x^2}(af_1+bf_2+cf_3+df_4+ef_5),\tag{5-34}$$

二阶精度

$$\begin{cases}a=\alpha+2+e, & b=-(2\alpha+5+4e),\\ c=\alpha+4+6e, & d=-(1+4e);\end{cases}\tag{5-35}$$

三阶精度

$$\begin{cases}a=\dfrac{11\alpha+35}{12},b=-\dfrac{5\alpha+26}{3},c=-\dfrac{\alpha+19}{2},\\ d=\dfrac{\alpha-14}{3}, \quad e=\dfrac{11-\alpha}{12}.\end{cases}\tag{5-36}$$

最近马延文等又提出了任意双精度的格式.首先讨论一阶导数逼近的对称差分.一阶导数具有四阶精度的差分为

$$\frac{1}{6}F_{j+1}+\frac{2}{3}F_j+\frac{1}{6}F_{j-1}=\delta_x^0 f_j.\tag{5-37}$$

若引进记号 $f^{(1)}=\Delta x\dfrac{\mathrm{d}f}{\mathrm{d}x},f^{(2)}=\Delta x^2\dfrac{\mathrm{d}^2f}{\mathrm{d}x^2},f^{(3)}=\Delta x^3\dfrac{\mathrm{d}^3f}{\mathrm{d}x^3}$，则(5-37)式可以改写为

$$\delta_x^0 f_j=\Delta x\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)_j+\frac{\Delta x^3}{3!}\left(\frac{\mathrm{d}^3f}{\mathrm{d}x^3}\right)_j+\cdots+\frac{1}{(2N-1)!}\left(\frac{\mathrm{d}^{2N-1}f}{\mathrm{d}x^{2N-1}}\right)_j,$$

略去高阶项即

$$\delta_x^0 f_j=f_j^{(1)}+\frac{1}{3!}f_j^{(3)}+\cdots+\frac{1}{(2N-1)!}f^{(2N-1)}.\tag{5-38}$$

另有辅助式

$$\frac{1}{2}[f_{j-1}^{(2n-1)}-2f_j^{(2n-1)}+f_{j+1}^{(un-1)}]-\frac{1}{2!}f_j^{(2n-1)}+\cdots+\frac{1}{[2(N-n)]!}f_j^{(2n-1)}=0,$$

$$n=1,2,\cdots,n-1,\tag{5-39}$$

它是由泰勒展开式得到的(当然也可以有别的取法).记

$$\begin{aligned} \boldsymbol{F} &= (f^{(1)}, f^{(3)}, \cdots, f^{(2n-1)})^{\mathrm{T}} \\ \boldsymbol{\theta} &= (1, 0, \cdots, 0) \end{aligned} \tag{5-40}$$

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

于是(5-38)和(5-39)式可以写成如下形式：

$$-\frac{1}{2}\boldsymbol{L}\boldsymbol{F}_{j-1} + (\boldsymbol{L}+\boldsymbol{A})\boldsymbol{F}_j - \frac{1}{2}\boldsymbol{L}\boldsymbol{F}_{j+1} = \delta_x^0 f_j \boldsymbol{e}_1, \tag{5-41}$$

其中 $\boldsymbol{A}$ 为

$$\boldsymbol{A} = \begin{bmatrix} \frac{1}{1!} & \frac{1}{3!} & \frac{1}{5!} & \cdots & \frac{1}{(2N-1)!} \\ 0 & \frac{1}{2!} & \frac{1}{4!} & \cdots & \frac{1}{[2(N-1)]!} \\ 0 & 0 & \frac{1}{2!} & \cdots & \frac{1}{[2(N-1)]!} \\ 0 & 0 & 0 & \cdots & \frac{1}{2!} \end{bmatrix}. \tag{5-42}$$

类似地有 $-\frac{1}{2}\boldsymbol{L}\boldsymbol{S}_{j-1} + (\boldsymbol{L}+\boldsymbol{B})\boldsymbol{S}_j - \frac{1}{2}\boldsymbol{L}\boldsymbol{S}_{j+1} = \frac{1}{2}\delta_x^2 f_j \boldsymbol{e}_1$, (5-43)

其中

$$\boldsymbol{B} = \begin{bmatrix} \frac{1}{2!} & \frac{1}{4!} & \frac{1}{6!} & \cdots & \frac{1}{(2N)!} \\ 0 & \frac{1}{2!} & \frac{1}{4!} & \cdots & \frac{1}{[2(N-1)]!} \\ 0 & 0 & \frac{1}{2!} & \cdots & \frac{1}{[2(N-2)]!} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{2!} \end{bmatrix}, \tag{5-44}$$

$$\boldsymbol{S} = (f^{(2)}, f^{(4)}, \cdots, f^{(2N)})^{\mathrm{T}}. \tag{5-45}$$

以上(5-39)、(5-41)、(5-43)式即为格式，可以用来计算超高精度的一、二阶导数. 在以上计算中看到 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{L}$ 与坐标方向无关，与网络点无关，所以矩阵计算的一些准备工作可以预先作好，避免重复计算. 但是又注意到 $\boldsymbol{A}$、$\boldsymbol{B}$ 都是刚性很大的矩阵，计算的精度是保证的($\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{L}$ 均为 $N\times N$ 方阵). 在边界点，则有

$$\delta_x^+ f_1 = \Delta x\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right) + \frac{\Delta x^2}{2!}\left(\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\right) + \cdots + \frac{1}{(2N-2)!}\frac{\mathrm{d}^{2N-2}f}{\mathrm{d}x^{2N-2}}, \tag{5-46}$$

即

$$f_2^{(0)} - f_1^{(0)} = f_1^{(1)} + \frac{1}{2!}f_1^{(n+2)} + \cdots + \frac{1}{(2N-n-2)!}f_1^{(2N-2)}, \tag{5-47}$$

$$f_2^{\langle n\rangle} - f_1^{\langle n\rangle} - \left(f_1^{\langle n+1\rangle} + \frac{1}{2!}f_1^{\langle n+2\rangle} + \cdots + \frac{1}{(2N-n-2)!}f_1^{\langle 2N-2\rangle}\right) = 0, \quad (5\text{-}48)$$

或写成

$$(\boldsymbol{L} + \boldsymbol{D})\widetilde{\boldsymbol{F}}_1 - \widetilde{\boldsymbol{F}}_2 = \delta_x^+ f_1 \widetilde{\boldsymbol{e}}_1, \quad (5\text{-}49)$$

其中

$$\widetilde{\boldsymbol{L}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \boldsymbol{D} = \begin{bmatrix} \frac{1}{1!} & \frac{1}{2!} & \cdots & \frac{1}{(2N-2)!} \\ 0 & \frac{1}{1!} & \cdots & \frac{1}{(2N-3)!} \\ 0 & 0 & \cdots & \frac{1}{(2N-4)!} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{1!} \end{bmatrix},$$

$\widetilde{\boldsymbol{F}} = (f_1^{\langle 1\rangle} f_1^{\langle 2\rangle} \cdots f_1^{\langle 2N-2\rangle})$, $\widetilde{\boldsymbol{e}}_1 = (1,0,\cdots,0)^T$, $\widetilde{\boldsymbol{L}}$、$\boldsymbol{D}$ 为 $(2N-2)\times(2N-2)$ 方阵. 在 N 点有类似的公式

$$-(\widetilde{\boldsymbol{L}} + \boldsymbol{D})\widetilde{\boldsymbol{F}}_{NJ} + \widetilde{\boldsymbol{F}}_{NJ-1} = \delta_x^- f_{NJ} \widetilde{\boldsymbol{e}}_1. \quad (5\text{-}50)$$

在有些情况下,如已知 $f_1^{\langle 1\rangle}$ 或 $f_{NJ}^{\langle 1\rangle}$,则方程组中第一式可以去掉,代之以边界条件.

在实际计算中,人们可以发现,在变化急剧处所求得的导数会出现振荡,和实际问题不符合,如接近阶跃函数的离散点,求导时,在阶跃附近的导数就会出现正负交替的不合理现象. 在这种情况下必须采用限制器,这一点是高阶精度格式尚未解决的问题. 在计算有激波等情况的流动时必须和 *TDV* 或 *ENO* 结合.

参考文献

1　苏铭德,黄素逸. 计算流体力学基础. 北京:清华大学出版社,1997.
2　水鸿寿. 一维流体力学差分方法. 北京:国防工业出版社,1998.
3　朱自强,吴子牛,李津等. 应用计算流体力学. 北京:北京航空航天大学出版社,1998.
4　陶文铨. 数值传热学. 西安:西安交通大学出版社,1988.
5　Toro E. Riemann Solvers and numerical methods for phind dynamics. New York: Springer Verlager, 1997.
6　Xu K. Gas-keretic schemes for unsteady Compressible phow simulation. Von Kárman Institute for phind Dynamics, Lechore Senies 1998.
7　Harten A. On the symmetric form of systems of conservation laws with entropy J of Comp Phys, 1983, 49: 151 ~ 164
8　Harton A, Engqmist B, Osher, S et al. Uniformly high order accurate essentially non-oscillatary scleme Ⅲ. J of Comp Phys, 1983, 49: 231 ~ 303.

9　张涵信.无波动、无自由参数的耗散差分格式.空气动力学学报,1988 6(2):143 ~ 165
10　付德重,马延文.迎风紧致格式及多尺度物理问题的直接数值模拟.第七届全国计算流体力学会议论文集,1994.

平衡态分布函数矩

(A)

平衡态分布函数即麦克斯韦分布函数 g.一维时

$$g = \rho\left(\frac{\lambda}{\pi}\right)^{\frac{k+1}{2}} e^{-\lambda[(u-U)^2+\xi^2]}.$$

引入记号

$$\rho\langle\cdot\rangle = \int_{-\infty}^{\infty}(\cdot)g\,\mathrm{d}u\,\mathrm{d}\xi,$$

不难看出
$$\langle u^n\xi^l\rangle = \langle u^n\rangle\langle\xi^l\rangle$$
当 l 是奇数时有 $\langle\xi^l\rangle = 0$.因为 g 关于 ξ 是对称函数,n 是一整数,计算可得

$$\langle\xi^2\rangle = \frac{K}{2\lambda},$$

$$\langle\xi^4\rangle = \left(\frac{3K}{4\lambda^2} + \frac{K(K-1)}{4\lambda^2}\right).$$

另有

$$\langle u^0\rangle = 1, \langle u^1\rangle = U, \langle u^2\rangle = U^2 + \frac{1}{2\lambda},$$

$$\langle u^3\rangle = U^3 + 1.5\frac{U}{\lambda}, \langle u^4\rangle = U^4 + \frac{3U^2}{\lambda} + \frac{0.75}{\lambda^2},$$

$$\langle u^5\rangle = U^5 + 5\frac{U^3}{\lambda} + 3.75\frac{U}{\lambda^2}, \langle u^6\rangle = U^6 + 7.5\frac{U^4}{\lambda} + 11.25\frac{U^2}{\lambda^2} + \frac{1.875}{\lambda^3},$$

一般递推公式为

$$\langle u^{n+2}\rangle = U\langle u^{n+1}\rangle + \frac{n+1}{2\lambda}\langle u^n\rangle.$$

再定义

$$\rho\langle\cdot\rangle = \int_{-\infty}^{\infty}(\cdot)g\,\mathrm{d}u\,\mathrm{d}\xi,$$

则有

$$\langle u^0\rangle_{u>0} = \frac{1}{2}\mathit{erfc}(-\sqrt{\lambda}U),$$

$$\langle u^1\rangle_{u>0} = \langle u^0\rangle_{u>0} + \frac{1}{2}\frac{e^{-\lambda u^2}}{\sqrt{\pi\lambda}}.$$

一般地

$$\langle u^{n+2}\rangle_{u>0} = U\langle u^{n+1}\rangle_{u>0} + \frac{n+1}{2\lambda}\langle u^n\rangle_{u>0}.$$

类似地

$$\langle u^1\rangle_{u>0}=\langle u^0\rangle_{u>0}+\frac{1}{2}\frac{e^{-\lambda u^2}}{\sqrt{\pi\lambda}},$$

$$\langle u^1\rangle_{u>0}=\langle u^0\rangle_{u<0}+\frac{1}{2}\frac{e^{-\lambda u^2}}{\sqrt{\pi\lambda}}.$$

一般地

$$\langle u^{n+2}\rangle_{u>0}=U\langle u^{n+1}\rangle_{u>0}+\frac{n+1}{2\lambda}\langle u^n\rangle_{u>0}.$$

另外三维时有

$$\langle u^m v^n w^k \xi^l\rangle=\langle u^m\rangle\langle v^n\rangle\langle w^k\rangle\langle \xi^l\rangle.$$

(B)

$$\beta F_{j+2}+\alpha F_{j+1}+F_{j-1}+\beta F_{j-2}=-\frac{c}{6\Delta x}f_{j-3}-\frac{b}{4\Delta x}f_{j-2}-\frac{a}{2\Delta x}f_{j-1}+\frac{a}{2\Delta x}f_{j+2}+\frac{c}{6\Delta x}f_{j+3}.$$

(1) $\beta=0, a=\frac{2}{3}(\alpha+2), b=\frac{1}{3}(4\alpha-1)$.

(2) $\alpha=\frac{1}{3}, \beta=0, a=\frac{14}{9}, b=\frac{1}{9}, c=0$.

(3) $\alpha=\frac{3}{8}, \beta=0, a=\frac{1}{6}(\alpha+9), b=\frac{1}{15}(32\alpha-9), c=\frac{1}{10}(-3\alpha+1)$.

(4) $a=\frac{1}{3}(4+2\alpha-16\beta+5c), b=\frac{1}{3}(-1+4\alpha+22\beta-8c)$.

(5) $a=\frac{1}{6}(9+\alpha-20\beta), b=\frac{1}{15}(-9+32\alpha+62\beta), c=\frac{1}{10}(1-3\alpha+12\beta)$.

(6) $\beta=\frac{1}{12}(-1+3\alpha), a=\frac{2}{9}(8-3\alpha), b=\frac{1}{18}(-17+57\alpha), c=0$.

(7) $\alpha=\frac{4}{9}, \beta=\frac{1}{36}, a=\frac{40}{27}, b=\frac{25}{54}, c=0$.

(8) $\beta=\frac{1}{20}(-3+8\alpha), a=\frac{1}{6}(12-7\alpha), b=\frac{1}{150}(568\alpha-183), c=\frac{1}{50}(9\alpha-4)$.

(9) $\alpha=\frac{1}{2}, \beta=\frac{1}{20}, a=\frac{17}{12}, b=\frac{101}{150}, c=\frac{1}{100}$.

其它紧致格式:

$\frac{1}{6}F_{j+1}+\frac{2}{3}F_j+\frac{1}{6}F_{j-1}=\frac{1}{2\Delta x}f_{j+1}-\frac{1}{2\Delta x}f_{j-1}$,四阶

$\frac{2}{3}F_j^+ +\frac{1}{3}F_{j-1}^+=\frac{f_{j+1}+4f_j-5f_{j-1}}{6\Delta x}$,三阶

$\frac{1}{3}F_{j+1}^- +\frac{2}{3}F_j^-=\frac{5f_{j+1}-4f_j-f_{j-1}}{6\Delta x}$,三阶

$\frac{3}{5}F_j^\pm+\frac{2}{5}F_{j\mp1}^\pm=\frac{1}{60}\delta_x^\mp(-f_{j\pm2}+11f_{j\pm1}+47f_j+3f_{j\mp2}$,五阶

其中 $\delta_x^{\mp} g_j = \mp(g_j - g_{j\pm 1})/\Delta x$.

二阶导数:

$$\beta s_{j-2} + \alpha s_{j-1} + s_j + \alpha s_{j+1} + \beta s_{j+2}$$
$$= \frac{c}{9\Delta x^2} f_{j-3} + \frac{b}{4\Delta x^2} f_{j-2} + \frac{a}{\Delta x^2} f_{j-1} - 2\left(\frac{c}{9\Delta x^2} + \frac{b}{4\Delta x^2} + \frac{a}{\Delta x^2}\right) f_i$$
$$+ \frac{a}{\Delta x^2} f_{j+1} + \frac{b}{4\Delta x^2} f_{j+2} + \frac{c}{9\Delta x^2} f_{j+3}.$$

(1) $\beta = 0, c = 0, a = \frac{4}{3}(1-\alpha), b = \frac{1}{3}(-1+10\alpha)$,四阶

(2) $\alpha = \frac{2}{11}, \beta = 0, a = \frac{12}{11}, b = \frac{3}{11}, c = 0$,六阶

(3) $a = \frac{1}{3}(4 - 4\alpha - 40\beta + 5c), b = \frac{1}{3}(-1 + 10\alpha + 46\beta - 8c)$,三参数四阶

(4) $a = \frac{1}{4}(6 - 9\alpha - 12\beta), b = \frac{1}{5}(-3 + 24\alpha - 6\beta), c = \frac{1}{20}(2 - 11\alpha + 124\beta)$
二参数六阶

(5) $\beta = \frac{38\alpha - 9}{214}, a = \frac{696 - 1191\alpha}{428}, b = \frac{2454\alpha - 294}{535}, c = \frac{1179\alpha - 344}{214}$,单参数八阶

(6) $\alpha = \frac{344}{1179}, c = 0, \beta, a, b$ 用(5)的公式.

(7) $\beta = \frac{43}{1798}, \alpha = \frac{334}{899}, a = \frac{1065}{1798}, b = \frac{1038}{899}, c = \frac{79}{1798}$,最高阶

其它 $\frac{1}{12}s_{j+1} + \frac{5}{6}s_j + \frac{1}{12}s_{j-1} = \frac{1}{\Delta x^2}(f_{j+1} - 2f_j + f_{j-1})$,四阶

高精度格式还有一些其它的构造方法,如解析离散法,一致逼近等,这些都正在发展中.高精度格式本身也有一些问题,正在研究和发展之中.

·计算机数学卷·

第 5 篇

多重网格法

编　者　顾丽珍
审校者　曾　实

目　　录

引　　言

多重网格法(multigrid method)是由前苏联人Fedorenk在1964年提出的.20世纪70年代,A.Brandt和W.Hackbush重新认识多重网格法的效率.80年代以后,多重网格学科的基本核心,包括基本原则、传统应用及理论都已建立,出现了一系列的多重网格法的文章和专著,其中有代表性的是参考文献[5～12].

多重网格法是一种"近乎最优"的特殊的迭代途径.在偏微分方程数值解中,当离散网格步长 h 变小时,它的收敛速度并不减慢,而经典的迭代法随 h 的变小而变慢.因此,多重网格法达到问题预定精度所需的计算工作量仅与未知量的个数 N 成正比,即 $O(N)$,比一般迭代法有效得多.

多重网格法广泛应用于微分方程和积分方程的数值解,特别是椭圆型偏微分方程.除此之外,还可应用于求解抛物型问题和其它依赖于时间的问题、本征值问题、分歧问题、非线性问题和直接用于无几何意义的代数方程组(称为代数多重网格法);在向量机或并行计算机上使用多重网格法更有效.因此,多重网格法是一种通用的非常有效的特殊类型的迭代法,已相当成功地应用于流体计算、结构力学计算和高度非线性的半导体器件模拟计算等领域.

本篇侧重于介绍多重网格法的基本原理、基本要素、多重网格算法,有关收敛性结论和计算机上的执行性能.

1　多重网格法基本原理

1.1　模　　型

要求解的连续问题是

$$Lu = f, \tag{1-1}$$

其中L是一个微分算子或积分算子或泛函求极值算子.

例1　两点边值问题:

$$\begin{cases} -u''(x) + \sigma u(x) = f(x), & 0 < x < 1, \quad \sigma \geqslant 0, \\ u(0) = u(1) = 0. \end{cases} \tag{1-2}$$

则 $L = -\dfrac{d^2}{dx^2} + \sigma$.

例2　二维泊松方程第一边值问题:

$$\begin{cases} -(u_{xx} + u_{yy}) = f(x,y), & (x,y) \in \Omega = \{(x,y) \mid 0 < x,y < 1,\} \\ u = 0, & (x,y) \in \partial\Omega. \end{cases} \tag{1-3}$$

则

$$\mathrm{L} = -\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right).$$

不论用有限差分法还是用有限元法求解这两个问题，首先对解域进行剖分，然后对微分算子进行离散，得到离散后的方程组

$$\mathbf{L}_h \boldsymbol{u}_h = \boldsymbol{f}_h. \tag{1-4}$$

对例 1 的解域[0,1]，均匀剖分成 N 等分，步长 $h = 1/N$，对微分算子用中心差分离散，得如下的离散方程组：

$$\begin{cases} \dfrac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} + \sigma u_j = f(x_j), & 1 \leqslant j \leqslant N-1, \\ u_0 = u_N = 0, \end{cases} \tag{1-5}$$

则有

$$\mathbf{L}_h = \frac{1}{h^2}\begin{bmatrix} 2+\sigma h^2 & -1 & & & \\ -1 & 2+\sigma h^2 & -1 & & \\ & & \ddots & \ddots & \ddots \\ & & -1 & 2+\sigma h^2 & -1 \\ & & & -1 & 2+\sigma h^2 \end{bmatrix}, \tag{1-6}$$

其中 $\mathbf{L}_h \in \mathbf{R}^{(N-1)\times(N-1)}$，$\boldsymbol{u}_h$、$\boldsymbol{f}_h \in \mathbf{R}^{N-1}$.

对例 2，若其解域为方形，I 和 J 分别为 x 方向和 y 方向的等分数，则 $h_x = 1/I$，$h_y = 1/J$. 用 Ω_h 表示该二维网格点的集合，$u(x_i, y_j)$ 表示微分方程的精确解，u_{ij} 表示差分方程精确解. 用五点差分格式去离散方程(1-3)得到差分方程组

$$\begin{cases} \dfrac{-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}}{h_x^2} + \dfrac{-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}}{h_y^2} = f_{i,j}, & 1 \leqslant i \leqslant I-1, 1 \leqslant j \leqslant J-1, \\ u_{i,j} = 0, & i = 0, I(0 \leqslant j \leqslant J), j = 0, J(0 \leqslant i \leqslant I). \end{cases} \tag{1-7}$$

若 $h_x = h_y = h$，按字典顺序排列 $(I-1)\times(J-1)$ 个未知量，得

$$\mathbf{L}_h = \frac{1}{h^2}\begin{bmatrix} A & -I & & & \\ -I & A & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & A & -I \\ & & & -I & A \end{bmatrix}, \tag{1-8}$$

其中 A、$I \in \mathbf{R}^{(I-1)\times(J-1)}$，$A$ 为弱对角占优的三对角阵，I 为对角阵，L_h 为 $(I-1)\times(J-1)$ 阶的大型稀疏矩阵.

1.2 多重网格法思想

对于离散后的线性代数方程组(1-4)，传统的迭代解法是在确定的一种网格 Ω_h 上采用某种迭代解法. 例如雅可比迭代法、高斯 - 塞德尔迭代法(简称 G-S 迭代)、逐次超松弛迭代(SOR)法、隐式交替方向(ADI)法以及它们的改进技术. 而多

重网格法恰恰不是在一种确定的网格上求解，而采用不同等级的均匀网格剖分．

假定有一组网格 $\Omega_{h_0},\Omega_{h_1},\cdots,\Omega_{h_M}$，称为网格序列 $\Omega_k(k=0,1,\cdots,M)$．随着 k 的增加，差分网格愈来愈细，这些网格都用同一种方式剖分得到，都逼近同一个区域 Ω．我们感兴趣的是要求解 Ω_M 上的差分方程组

$$\mathbf{L}_M\boldsymbol{u}_M=\boldsymbol{f}_M.\tag{1-9}$$

称在 Ω_k 上解 $\mathbf{L}_k\boldsymbol{u}_k=\boldsymbol{f}_k$ 的问题为 Ω_k 问题，称在 Ω_M 上求解差分方程(1-9)的问题为 Ω_M 问题．

对于线性问题，令 $\bar{\boldsymbol{u}}_M$ 为 Ω_M 问题的近似解，$\boldsymbol{u}_M$ 为精确解，则有

$$\boldsymbol{u}_M=\bar{\boldsymbol{u}}_M+\boldsymbol{v}_M,$$

其中 $\boldsymbol{v}_M$ 称为修正量．代入方程(1-9)后，得

$$\mathbf{L}_M\boldsymbol{v}_M=\boldsymbol{d}_M.$$

称上式为**亏损方程**，其中

$$\boldsymbol{d}_M=\boldsymbol{f}_M-\mathbf{L}_M\bar{\boldsymbol{u}}_M$$

称为**亏损量**．

线性多重网格法是一种特殊的迭代过程，它把 Ω_M 上求解方程(1-9)与在粗网 Ω_k 上求解亏损方程

$$\mathbf{L}_k\boldsymbol{v}_k=\boldsymbol{d}_k\tag{1-10}$$

相结合，而求解 Ω_k 上的亏损方程又把在 Ω_k 上迭代(1-10)与求解 Ω_{k-1} 上的亏损方程相结合．这样一种巧妙的结合构成一种收敛速度很快的迭代方法，这就是多重网格法的基本思想．

1.3　双网格方法

双网格方法(two-grid method)是多重网格法的基础．首先叙述仅附加一种网格的双网格方法，然后再推广到一系列网格的多重网格方法．

双网格方法用两种网格，一种用 Ω_H 表示，另一种用 Ω_h 表示．网格步长分别为 H 和 h．通常 $H=2h$．在 Ω_H 和 Ω_h 上的差分算子分别为 $\mathbf{L}_H$ 和 $\mathbf{L}_h$，且设 $\mathbf{L}_H^{-1}$ 和 $\mathbf{L}_h^{-1}$ 存在．

现在的任务是在 Ω_h 上求解方程组．

$$\mathbf{L}_h\boldsymbol{u}_h=\boldsymbol{f}_h.\tag{1-11}$$

粗细两种网格上的值需要转移，下面定义由粗网到细网及由细网到粗网上网格函数的转移算子．

插值(interpolation)**算子或延拓**(prolongation)**算子** $\mathbf{I}_H^h$ 表示粗网到细网上值的转移．如一维线性插值

$$\mathbf{I}_H^h \stackrel{\text{def}}{=} \frac{1}{2}]1 \quad 2 \quad 1[_H^h \quad 或 \quad \mathbf{I}_H^h = \frac{1}{2}\begin{bmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & 2 \\ & & & 1 \end{bmatrix} \tag{1-12}$$

(1-12)式表示粗网上的值按此权系数分配到邻近点的细网上去，

$$\mathbf{L}_H^h u_H = u_h,$$

即

$$\begin{cases} u_{2j}^h = u_j^H, \\ u_{2j+1}^h = \dfrac{u_j^H + u_{j+1}^H}{2}, \end{cases} \qquad 0 \leqslant j \leqslant \frac{N}{2} - 1.$$

限制(restriction)**算子** $\mathbf{I}_h^H$ 表示由细网到粗网上值的转移.如一维限制算子

$$\mathbf{I}_h^H \stackrel{\text{def}}{=} \frac{1}{4}[1 \quad 2 \quad 1]_h^H \quad 或 \quad \mathbf{I}_h^H = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & & & \ddots & & \\ & & & & & 1 & 2 & 1 \end{bmatrix},$$

$$\mathbf{I}_h^H u_h = u_H, \tag{1-13}$$

即

$$u_j^H = \frac{u_{2j-1}^h + 2u_{2j}^h + u_{2j+1}^h}{4}, \qquad 1 \leqslant j \leqslant \frac{N}{2} - 1.$$

下面叙述用双网格方法求解方程(1-11)时,由已知 $u_h^{(n)}$ 计算 $u_h^{(n+1)}$ 的一个迭代步或一个循环.共分三个阶段:

(1) **前光滑**(pre-smoothing):

给定初值 $u_h^{(n)}$,选定一种松弛法,在 Ω_h 上对方程(1-11)作 ν_1($\nu_1 = 1$或2)次迭代(也称松弛)计算 $\overline{u}_h^{(n)}$

$$\overline{u}_h^{(n)} \stackrel{\text{def}}{=} S^{\nu_1}(u_h^{(n)}, L_h, f_h).$$

(2) **粗网修正**(coarse-grid correction,简称CGC):

1) 计算细网亏损量:$d_h^{(n)} = f_h - \mathbf{L}_h \overline{u}_h^{(n)}$,

2) 限制亏损量(由细网到粗网):$d_H^{(n)} = \mathbf{I}_h^H d_h^{(n)}$,

3) 在 Ω_H 上求 $\mathbf{L}_H v_H^{(n)} = d_H^{(n)}$ 的精确解 $v_H^{(n)}$, (1-14)

4) 插值修正量(由粗网到细网):$v_h^{(n)} = \mathbf{I}_H^h v_H^{(n)}$

5) 对 $\overline{u}_h^{(n)}$ 作修正: $\widehat{u}_h^{(n)} = \overline{u}_h^{(n)} + v_h^{(n)}$,

(3) **后光滑**(post-smoothing):

以 $\widehat{u}_h^{(n)}$ 为迭代的初始近似值,在 Ω_h 上对方程(1-11)作 ν_2($\nu_2 = 1$或2)次迭代

计算 $u_h^{(n+1)}$,

$$u_h^{(n+1)} \xlongequal{\text{def}} S^{\nu_2}(\widehat{u}_h^{(n)}, L_h, f_h).$$

为直观起见,用下列路径表示双网格的一个循环步的计算:

$$\begin{array}{lllll}
\Omega_h & u_h^{(n)} \xrightarrow{S^{\nu_1}} \bar{u}_h^{(n)} \to d_h^{(n)} = f_h - L_h\bar{u}_h^{(n)}, & & v_h^{(n)} \to \widehat{u}_h^{(n)} = \bar{u}_h^{(n)} + v_h^{(n)} \xrightarrow{S^{\nu_2}} u_h^{(n+1)} \\
& \quad I_h^H \downarrow & & \quad I_H^h \uparrow & \\
\Omega_H & \quad d_H^{(n)} & \longrightarrow & L_H v_H^{(n)} = d_H^{(n)} &
\end{array}$$

这过程示于图 1-1,图中 ∘、□、↘、↗ 分别表示松弛,求精确解和限制亏损量及插值修正量.

图 1-1

这样一种特殊迭代过程的迭代公式为

$$u_h^{(n+1)} = M_h^H u_h^{(n)} + N_h(f_h),$$

其双网格迭代矩阵为

$$M_h^H = S^{\nu_2} K_h^H S^{\nu_1}, \tag{1-15}$$

其中

$$K_h^H \xlongequal{\text{def}} I_h - I_H^h L_H^{-1} I_h^H L_h \tag{1-16}$$

是粗网修正的迭代矩阵.

1.4 多重网格法原理——一维模型问题分析

细网上松弛 S^{ν_1}、S^{ν_2} 和粗网修正是双网格方法的两大要素.若仅在细网上作松弛,如雅可比 ω 松弛、高斯-塞德尔松弛和 SOR 松弛,当网格步长 $h \to 0$ 时,收敛性变得很差.若单独用粗网修正,在迭代时,粗网修正的迭代矩阵 K_h^H 的谱半径 $\rho(K_h^H) \geqslant 1$,从而不收敛.如果把细网松弛和粗网修正相结合,则构成一种收敛速度很快的特殊迭代方法.下面以 1.1 节中的一维模型($\sigma = 0$)为例,对这两个要素作具体分析.

1.4.1 细网松弛

细网上方程 $Au = b$ 的定常迭代法为

$$u^{(n+1)} = Bu^{(n)} + g. \tag{1-17a}$$

将方程的系数矩阵 A 分解为 $A = D - L - U$,其中 D 为 A 的对角线元素构成的对角阵,$-L$ 为 A 的下三角部分构成的严格下三角阵,$-U$ 为 A 的上三角部分构成的严格上三角阵.当 B 分别取

(1) $B_J = D^{-1}(L + U)$,

(2) $B_{S\text{-}G} = (D - L)^{-1}U$,

(3) $B_{SOR} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$, $0 < \omega < 2$, (1-17b)

(4) $B_{J\omega} = (1 - \omega)I + \omega B_J = I - \omega D^{-1}A$, $0 < \omega < 1$

时,可分别得到雅可比迭代,G-S迭代,SOR迭代和雅可比-ω迭代法(又称为阻尼

雅可比迭代).迭代收敛的充分必要条件是 $\rho(\boldsymbol{B}) < 1$,迭代收敛的一种充分条件是 $\|\boldsymbol{B}\| < 1$. 称 $\rho(\boldsymbol{B})$ 为**渐近收敛因子**,称 $R(\boldsymbol{B}) = -\ln\rho(\boldsymbol{B})$ 为**渐近收敛率**,称 $\|\boldsymbol{B}\|$ 为**收缩数**.

对 $\sigma = 0$ 的一维离散模型(1-5)和(1-6)式,令 $\boldsymbol{A} = h^2 L_h, \boldsymbol{b} = h^2 f_h \quad h = 1/N$, 易求出 $\boldsymbol{A}$ 的第 k 个特征值和相应的第 k 个特征向量的第 j 个分量,它们分别为

$$\begin{cases} \lambda_k(\boldsymbol{A}) = 4\sin^2\dfrac{k\pi}{2N}, & 1 \leqslant k \leqslant N-1, \\ w_{k,j} = \sin\dfrac{jk\pi}{N}, & 1 \leqslant k \leqslant N-1, \end{cases} \tag{1-18}$$

其中 $w_{k,j}$ 也是矩阵 $\boldsymbol{A}$ 的傅里叶模.

雅可比迭代阵的特征值和特征向量的第 j 个分量分别为

$$\begin{cases} \lambda_k(\boldsymbol{B}_{J\omega}) = 1 - 2\sin^2\dfrac{k\pi}{2N}, & 1 \leqslant k \leqslant N-1, \\ w_{k,j} = \sin\dfrac{jk\pi}{N}, & 1 \leqslant k \leqslant N-1. \end{cases} \tag{1-19}$$

雅可比 -ω 迭代阵的特征值和特征向量的第 j 个分量分别为

$$\begin{cases} \lambda_k(\boldsymbol{B}_J) = 1 - \dfrac{\omega\lambda(\boldsymbol{A})}{2} = 1 - 2\omega\sin^2\dfrac{k\pi}{2N}, & 1 \leqslant k \leqslant N-1, \\ w_{k,j} = \sin\dfrac{jk\pi}{N}, & 1 \leqslant k \leqslant N-1. \end{cases} \tag{1-20}$$

G-S 迭代阵的特征值和特征向量的第 j 个分量分别为

$$\begin{cases} \lambda_k(B_{G-S}) = \cos^2\dfrac{k\pi}{N}, & 1 \leqslant k \leqslant N-1 \\ w_{k,j}^{G-S} = \cos^j\dfrac{k\pi}{N}\sin\dfrac{jk\pi}{N} = \lambda_k^{j/2}\sin\dfrac{jk\pi}{N}, & 1 \leqslant j \leqslant N, 1 \leqslant k \leqslant N-1. \end{cases} \tag{1-21}$$

下面先分析松弛对初始误差的光滑作用.设迭代的初始误差向量用 $\boldsymbol{A}$ 的 $N-1$ 个线性无关的特征向量表示为

$$\boldsymbol{e}^{(0)} = \sum_{k=1}^{N-1} c_k \boldsymbol{W}_k, c_k \in \mathbf{R}. \tag{1-22}$$

用(1-17)式作 n 次迭代后

$$\boldsymbol{e}^{(n)} = \boldsymbol{B}^n \boldsymbol{e}^{(0)} = \sum_{k=1}^{N-1} c_k \lambda_k^n(\boldsymbol{B}) \boldsymbol{W}_k, \quad c_k \in \mathbf{R}. \tag{1-23}$$

这说明经 n 次迭代后,初始误差的第 k 个模减小到 $\lambda_k^n(\boldsymbol{B})\boldsymbol{W}_k$. 把级数(1-23)的各项分成两部分,即高频分量和低频分量,示于图 1-2.其中 k 刚好给出 Ω_h 上"半波"的个数.所谓低频即在粗网 Ω_H 上能表现出来的那部分特征函数,而高频是在 Ω_H 上根本看不见的那部分特征函数.具体定义为

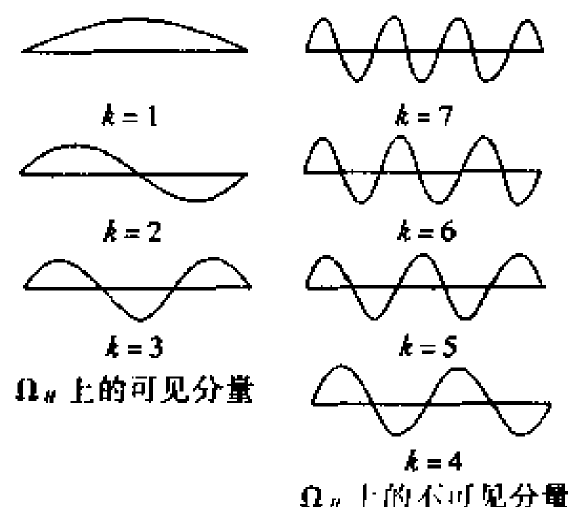


图 1-2

低频：　　$w_{k,j},\quad 1 \leqslant k < \dfrac{N}{2}$

高频：　　$w_{k,j},\quad \dfrac{N}{2} \leqslant k \leqslant N-1$

为了度量松弛方法对误差的光滑程度，定义

$$\mu(\boldsymbol{B}) = \max_{N/2 \leqslant k \leqslant N-1} |\lambda_k(\boldsymbol{B})|,$$

为松弛法的**光滑因子**（smoothing factor），即松弛后消除误差的高频振荡分量，使误差变光滑，但并不是显著地减小误差. 图 1-3 为松弛法典型的光滑误差特性.

松弛前

松弛后

低频：振幅稍减　　高频：振幅显著减少

图 1-3

对阻尼雅可比方法，选 ω 使 $\max\{\lambda_{N/2}(\boldsymbol{B}_{J\omega}), \lambda_N(\boldsymbol{B}_{J\omega})\}$ 达到最小，即

$$\lambda_{N/2}(\boldsymbol{B}_{J\omega}) = -\lambda_N(\boldsymbol{B}_{J\omega}),$$

从而求出最佳参数 $\omega = 2/3$. 图 1-4 表示了 $\omega = 1/3, 1/2, 2/3, 1$ 时 $\lambda_k(\boldsymbol{B}_{J\omega})$ 随 k 变化的曲线，显然，$\omega = 2/3$ 的阻尼雅可比法能最好地阻尼高频分量. 这时

$$\lambda_k(\boldsymbol{B}_{J2/3}) = 1 - \frac{4}{3}\sin^2\frac{k\pi}{2N}, \quad 1 \leqslant k \leqslant N-1. \tag{1-24}$$

当 $N/2 \leqslant k \leqslant N-1$ 时，$|\lambda_k(\boldsymbol{B}_{J2/3})| \leqslant 1/3$，即有光滑因子 $\mu(\boldsymbol{B}_{J2/3}) = 1/3$，可见对误差的高频分量有很好的光滑作用. 当 $0 < k < N/2, N = 8$ 时，$1/2 < |\lambda_k(\boldsymbol{B}_{J2/3})| < 1$，因此，对误差的低频分量衰减不大. 当 $\omega = 1$ 时，雅可比-ω 迭代法就是雅可比迭代法，它对误差的高频和低频分量的衰减均很慢，仅对 $N/2$ 附近的波阻尼快，因此雅可比方法对误差的高频分量光滑效应差，在多重网格法中一般不采用.

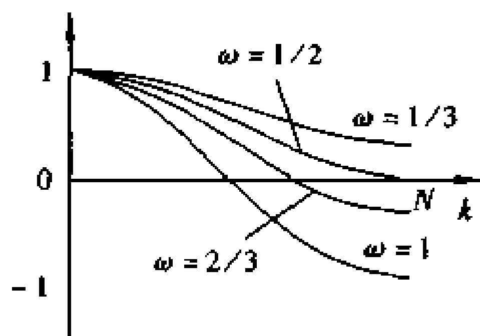


图 1-4

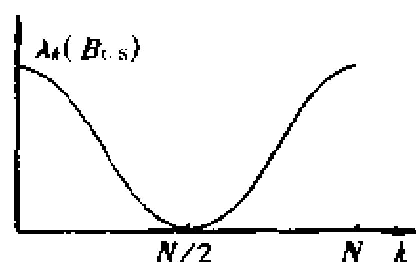


图 1-5

再分析 G-S 迭代法. 图 1-5 表示 $\lambda_k(\boldsymbol{B}_{G\text{-}S}) = \cos^2(k\pi/N)$ 随 k 变化的曲线，其特征向量

$$w_{k,j}^{G\text{-}S} = \lambda_k^{j/2}\sin\frac{jk\pi}{N} = \lambda_k^{j/2} w_{k,j},$$

则有

$$e^{(0)} = \sum_{k=1}^{N-1} c_k W_k^{G\text{-}S}, \quad e^{(n)} = B_{G\text{-}S} e^{(0)} = \sum_{k=1}^{N-1} c_k \lambda_k^{n+j/2}(B_{G\text{-}S}) W_k,$$

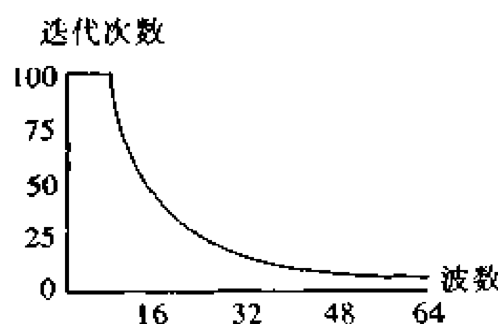


图 1-6

其中 W_k 为矩阵A 的特征向量.要想解析地说明 G-S 迭代法对误差的高频分量的快衰减是困难的,从数值试验结果进行分析,对 $N = 64$ 的一维模型,若以 W_k 作为初始猜测,则用 G-S 迭代将 W_k 减小到原来的 1/100 所需的迭代次数与波数 k 之间的关系曲线示如图 1-6 所示.由此可见,G-S 迭代法对低频误差衰减很慢,对高频的振荡误差衰减很快.

从以上分析可知,作几次阻尼雅可比迭代或 G-S 迭代均能有效地衰减误差的高频分量.振荡模一旦被消去,留下光滑模,再继续迭代已无效.弥补的办法之一是采用多重网格,特别是粗网修正.因此,有转向粗网计算的必要性.

1.4.2　粗网修正

先看谱半径

$$\rho(B_{J\omega}) = \max_k | 1 - 2\omega\sin^2\frac{k\pi}{2N} | \approx 1 - \frac{\omega\pi^2h^2}{2} = 1 - O(h^2),$$

$$\rho(B_J) = \max_k | 1 - 2\sin^2\frac{k\pi}{2N} | \approx 1 - \frac{\pi^2h^2}{2} = 1 - O(h^2),$$

$$\rho(B_{G\text{-}S}) = \max_k | \cos^2\frac{k\pi}{N} | = 1 - O(h^2),$$

同理有 $\rho(B_{SOR}) = 1 - O(h)$.从而在粗网上,由于网格变粗、h 变大,迭代谱半径均变小,收敛率增加,因此,转向粗网有加速收敛的作用.

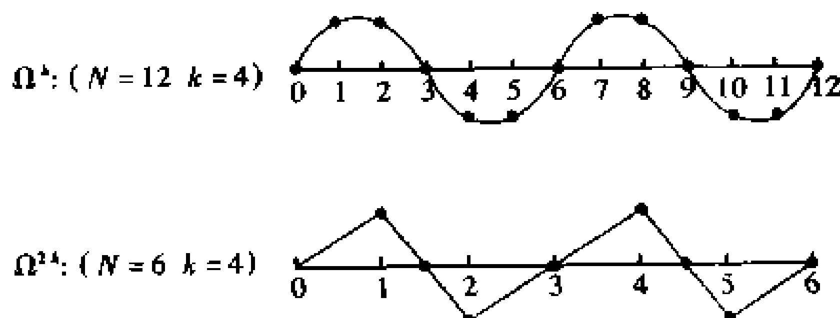


图 1-7

再看双网格法 CGC 中(2),从粗网转移到细网后的作用.从图 1-7 可见,$\Omega_h(N = 12)$ 上 $k = 4$ 的波是低频波,限制转移到粗网 $\Omega_{2h}(N = 6)$ 上是 $k = 4$ 的高频波.在粗网上松弛可将粗网上的高频模约化,即减小了细网上的低频模,比一直在细网上迭代消灭不了低频模更有效,使收敛速度加快.

但 Ω_{2h} 上仍会留下它的低频模.

对应于双网格法的CGC中(3),在多重网格法中,这一步不是精确地求解,而是迭代求解亏损方程.迭代同样只能光滑 Ω_{2h} 上的高频分量,同理只需迭代1—2次后便有转向更粗网格 Ω_{4h} 的必要.重复这个过程,继续转向 Ω_{8h}、Ω_{16h} 等等,直到最粗网格上的亏损量可精确地求解.这是多重网格法与双网格法不同之处.

在 CGC(3) 中,为何求解亏损方程 $\mathbf{L}_H v_H^{(n)} = d_H^{(n)}$ 而不去求解粗网方程 $\mathbf{L}_H u_H = f_H$?这是因为希望最终的亏损量是零向量,从而可用零向量作为迭代的初始量,这比起用 $u_H^{(n)} = \mathbf{I}_h^H u_h^{(n)}$ 作为初始向量去求解 $\mathbf{L}_H u_H = f_H$ 收敛速度快,也方便.

在 CGC(4) 和(5) 中,将粗网函数转移到细网及对细网解进行修正,这使细网解得到更好的近似值.

综上所说,多重网格法是把在细网上松弛消失误差中的高频振荡分量,使误差光滑化,和在粗网上低频分量容易收敛及在粗网上修正为细网提供更好的初始值等相结合,形成一种特殊的收敛快的迭代过程.

例 3　用多重网格法计算一维模型问题 $Au = 0(N = 48)$ 的解.取初始向量 $u_{h,j}^{(n)} = (\sin(12j\pi/N) + \sin(30j\pi/N))/2$,用 $\omega = 2/3$ 的阻尼雅可比松弛,前松弛 $\nu_1 = 3$,后松弛 $\nu_2 = 3$.

设 u 为一维问题的精确解向量,u_h 为细网上的近似解向量,u_{2h} 为粗网上的近似解向量,初始误差向量 $\| e \|_\infty = \| u - u_h \|_\infty = \| - u_h \| = 1.00$.

一个双网格迭代步的计算结果:

(1) 在细网上松弛3次后的误差为 $\| e_h \|_\infty = 0.261$,这时误差的振荡分量基本消去.

(2) 用完全加权算子转移亏损量到粗网,在 Ω_{2h} 上对 $A_{2h}e_{2h} = d_{2h}$ 松弛3次后,有误差 $\| e_{2h} \| = 0.0591$,为初始误差的6%.

(3) 粗网修正后,再作3次细网松弛,有误差 $\| e_h \|_\infty = 0.02$.

(4) 再把亏损量转移到粗网,又在 Ω_{2h} 上松弛3次后,有误差 $\| e_{2h} \| = 0.00977$.

1.4.3　双网格方法收敛性

定理 1　用双网格方法求解一维模型问题,选择 $\omega = 2/3$ 的阻尼雅可比松弛法,取 $\nu_2 = 0$,u_h 为细网上的近似解向量,其初始误差向量为 e_h,经双网格法一个循环步以后的误差为 $\bar{e}_h$,则有误差估计

$$\| \bar{e}_h \|_2^2 \leqslant \left(\frac{1}{2} + \frac{1}{3^{\nu_1}}\right) \| e_h \|_2^2.$$

当 $\nu_1 = 2$ 时,

$$\| \bar{e}_h \|_2 \leqslant 0.782 \| e_h \|_2 .$$

j 次双网格步后的误差为

$$\| \bar{e}_h^{(j)} \|_2 \leqslant 0.782^j \| e_h^{(0)} \|_2 .$$

因此,双网格方法以一个与 h 无关的收敛速率收敛于零.

双网格方法在实际计算中用得不多，但它是多重网格方法的理论基础，通过它阐述了多重网格法的基本原理.

2　线性多重网格法

多重网格法是在愈来愈粗的网格上递归地使用双网格方法. 仅在最粗的网格上精确地求解差分方程. 现在定义多重网格法所用的符号和算子：

网格步长序列　$h_k(k=0,1,\cdots,M)$，通常 $h_k=h_{k-1}/2=2^{-k}h$.

网格序列　$\Omega_k(k=0,1,\cdots,M)$.

差分算子序列　$\mathbf{L}_k(k=0,1,\cdots,M)$，设 L_k^{-1} 存在.

松弛算子序列　$\mathrm{S}^{\nu}(u_k,\mathbf{L}_k,f_k)$.

限制算子序列　$\mathbf{I}_k^{k-1}(k=1,2,\cdots,M):G(\Omega_k)\to G(\Omega_{k-1})$.

插值算子序列　$\mathbf{I}_{k-1}^{k}(k=1,2,\cdots,M):G(\Omega_{k-1})\to G(\Omega_k)$.

其中 $G(\Omega_k)$ 为在 Ω_k 上网格函数的线性空间.

下面叙述用多重网格法迭代求解方程组

$$\mathbf{L}_M u_M=f_M \tag{2-1}$$

的计算过程. 设已知 $u_M^{(n)}$，求 $u_M^{(n+1)}$，即

$$u_M^{(n+1)}\overset{\text{def}}{=\!=}\mathrm{MR}_M(u_M^{(n)},L_M,f_M), \tag{2-2}$$

其中 R = V 为 V 循环，R = W 为 W 循环. 仿照双网格方法，计算分为三个阶段：

(1) 前光滑：给定初值 $u_M^{(n)}$，在 Ω_M 上作 $\nu_1(\geqslant 1)$ 次松弛得

$$\bar{u}_M^{(n)}\overset{\text{def}}{=\!=}\mathrm{S}^{\nu_1}(u_M^{(n)},L_M,f_M).$$

(2) 粗网修正：

1) 计算亏损量 $d_M^{(n)}\overset{\text{def}}{=\!=}f_M-\mathbf{L}_M\bar{u}_M^{(n)}$.

2) 限制亏损量 $d_{M-1}^{(n)}\overset{\text{def}}{=\!=}\mathbf{I}_M^{M-1}d_M^{(n)}$.

3) 在 Ω_{M-1} 上近似求解亏损方程

$$\mathbf{L}_{M-1}v_{M-1}^{(n)}=d_{M-1}^{(n)}. \tag{2-3}$$

计算(2-3)式时，以零网格函数作为初始近似值，即 $v_{M-1}^{(n)}=0$，作 $M-1$ 网格法的 r 型迭代. $M-1$ 网格法用 $\Omega_{M-1},\Omega_{M-2},\cdots,\Omega_0$，即

$$v_{M-1}^{(n)}\overset{\text{def}}{=\!=}\mathrm{MR}_{M-1}(v_{M-1}^{(n)},L_{M-1},d_{M-1}).$$

在最粗的网格上精确求解

$$\mathbf{L}_0 v_0=d_0.$$

4) 插值　$v_M^{(n)}\overset{\text{def}}{=\!=}\mathbf{I}_{M-1}^{M}v_{M-1}^{(n)}$.

5) 修正 Ω_M 上的节点函数值 $\hat{u}_M^{(n)}=\bar{u}_M^{(n)}+v_M^{(n)}$.

(3) 后光滑：在 Ω_M 上作 $\nu_2(\geqslant 1)$ 次迭代得

$$u_M^{(n+1)} \xlongequal{\text{def}} S^{\nu_2}(\widehat{u}_M^{(n)}, \mathbf{L}_M, f_M).$$

上面粗网修正中的 r 也是循环参数，一般 $r = 1,2$ 或 3. 当 $r > 3$ 时多重网格法就不太有效了. 我们称 $r = 1$ 为 V 循环，$r = 2$ 为 W 循环，$r = 3$ 很少见.

用图 2-1 的符号分别表示 $M = 1,2,3$ 时，$r = 1,2,3$ 的多重网格法的一个循环(或一个迭代步)，如图 2-1 所示. 图 2-2 的流程图表示一个多重网格迭代步的计算过程，其中引入了一个开关参数 $C(K)$，$0 \le C(K) \le r$，以控制转向粗网和返回细网.

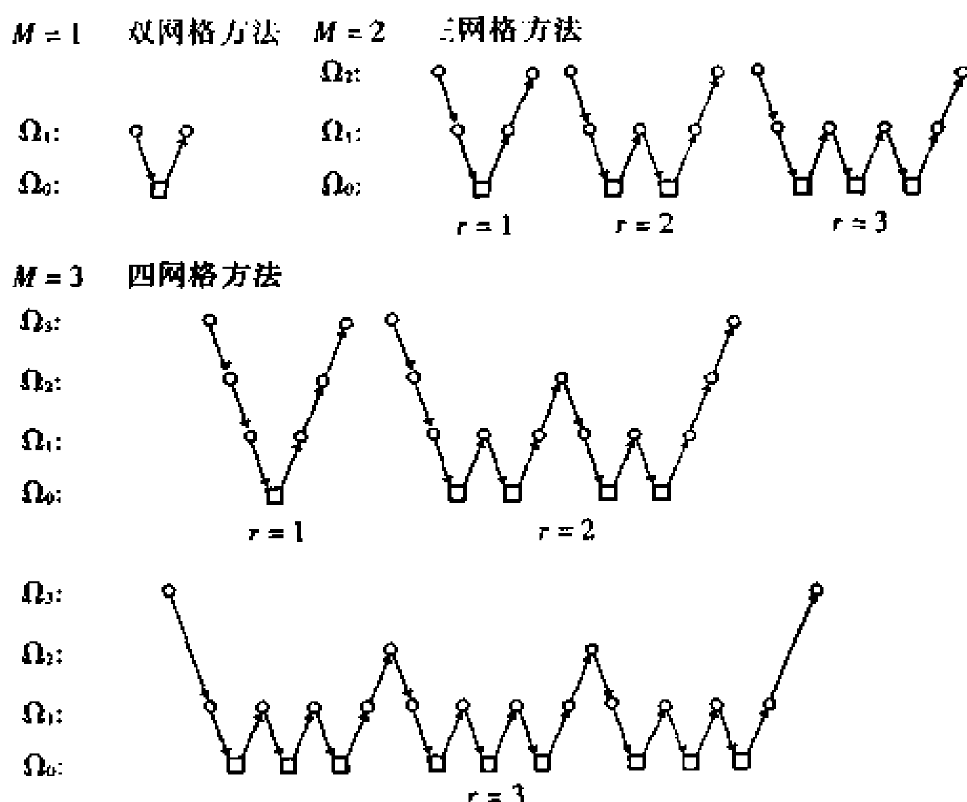


图 2-1

多重网格法的迭代公式是

$$u_M^{(n+1)} = \mathbf{M}_M u_M^{(n)} + \mathbf{N}_M(f_M),$$

其中

$$\mathbf{M}_1 \xlongequal{\text{def}} \mathbf{S}_1^{\nu_2}(\mathbf{I}_1 - \mathbf{I}_0^1 \mathbf{L}_0^{-1} \mathbf{I}_1^0 \mathbf{L}_1)\mathbf{S}_1^{\nu_1}$$

$$\mathbf{M}_k \xlongequal{\text{def}} \mathbf{S}_k^{\nu_2}(\mathbf{I}_k - \mathbf{I}_{k-1}^k(\mathbf{I}_{k-1} - A_{k-1}^r)\mathbf{L}_{k-1}^{-1}\mathbf{I}_k^{k-1}\mathbf{L}_k)\mathbf{S}_k^{\nu_1}, \quad k = 2,3,\cdots,M. \tag{2-4}$$

而 Ω_{M-1} 和 Ω_M 的双网格算子为

$$\mathbf{M}_M^{M-1} \xlongequal{\text{def}} \mathbf{S}_M^{\nu_2}\ (\mathbf{I}_M - \mathbf{I}_{M-1}^M \mathbf{L}_{M-1}^{-1} \mathbf{I}_M^{M-1} \mathbf{L}_M)\mathbf{S}_M^{\nu_1}. \tag{2-5}$$

(2-4) 式与(2-5) 式的差别在于用$(\mathbf{I}_{M-1} - A_{M-1}^r)\mathbf{L}_{M-1}^{-1}$ 代替(2-5) 式中的 $\mathbf{L}_{M-1}^{-1}$，这说明用 $\Omega_{M-1}, \Omega_{M-2}, \cdots, \Omega_0$ 的 $M - 1$ 网格 r 型多重网格迭代去近似求解

$$\mathbf{L}_{M-1} v_{M-1}^{(n)} = d_{M-1}^{(n)}.$$

以上是线性多重网格法的计算步骤. 关于多重网格法的收敛性，一般来说，若双网格方法收敛很好，则相应的 $r = 2$ 的多重网格法亦收敛得很好.

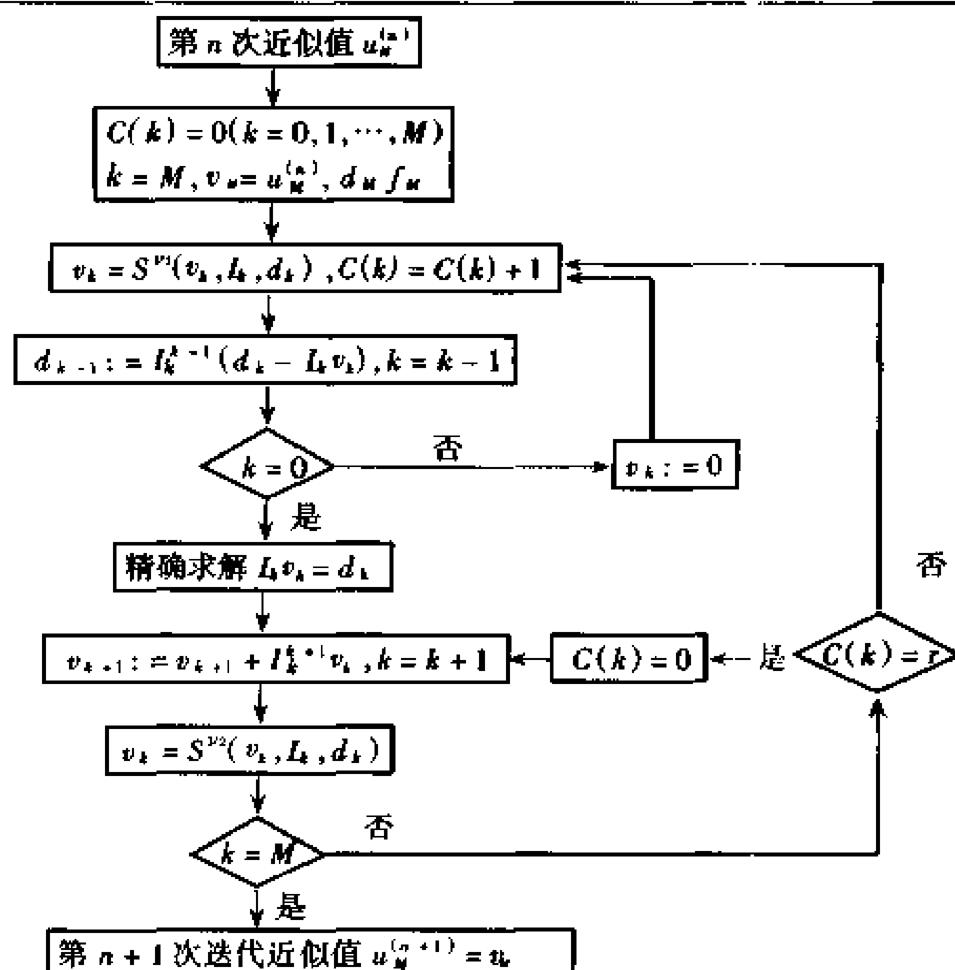


图 2-2

3　完整的多重网格法

多重网格法与完整网格技巧相结合，或者说与套迭代技术相结合，称完整的多重网格法(full multigrid，简称 FMG 方法)，FMG 方法效果更明显．所谓套迭代就是粗网为细网提供良好的初值．这是古典迭代方法(例如 SOR 方法)不可能做到的．

FMG 方法的计算步骤和流程图分别示于图 3-1 和图 3-2 中．在图 3-2 中，INT 表

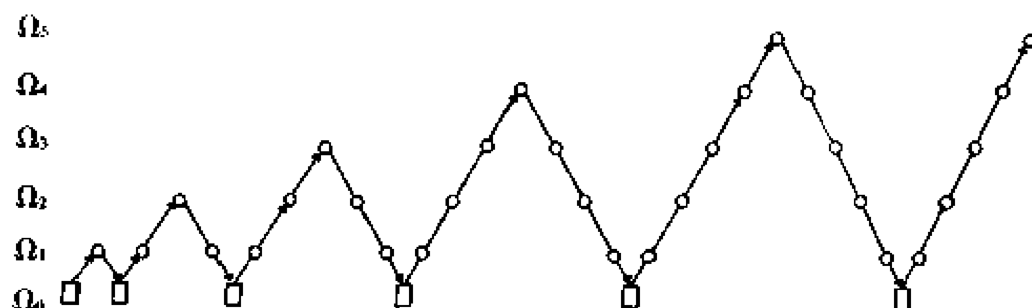


图 3-1

示插值,MGI 表示在 $\Omega_k(k=0,1,\cdots,M)$ 上用多重网格 r 型迭代求解方程 $\mathbf{L}_k u_k=f_k(k=0,1,\cdots,M)$. 当 $r=1$ 时为 V 循环,称为 **FMV 方法**, $r=2$ 时为 W 循环,称为 **FMW 方法**. 在图 3-2 中,当 $M=6, r=1$ 时为图 3-1 所示的 FMV 过程.

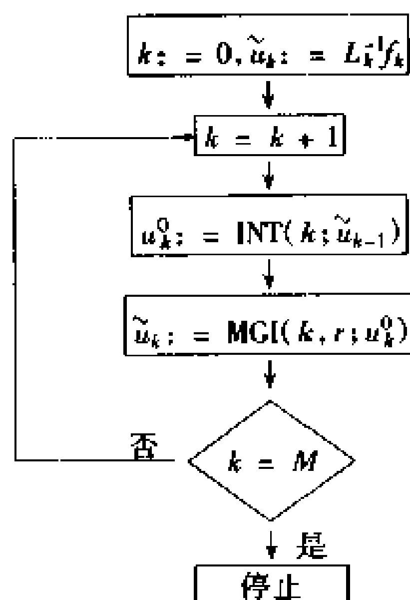


图 3-2

由(2-2)式,一个多重网格法的 V 循环步为

$$u_M^{(n+1)}:=\mathrm{MV}_M(u_M^{(n)},L_M,f_M).$$

一个 FMV 的循环步定义为

$$u_M^{(n+1)}:=\mathrm{FMV}_M(u_M^{(n)},L_M,f_M)$$

其计算步骤如下:

将 $f_M,f_{M-1},f_{M-2}\cdots$ 及 $u_M,u_{M-1},u_{M-2},\cdots$ 初始化为零. 在最粗网格上松弛或直接求解方程 $A_0u_0=f_0$,

$$\begin{aligned}
&\cdots\cdots\\
&u_{M-2}=u_{M-2}+\mathbf{I}_{M-3}^{M-2}u_{M-3},\\
&u_{M-2}=\mathrm{MV}_{M-2}(u_{M-2},\mathbf{L}_{M-2},f_{M-2}),\\
&u_{M-1}=u_{M-1}+\mathbf{I}_{M-2}^{M-1}u_{M-2},\\
&u_{M-1}=\mathrm{MV}_{M-1}(u_{M-1},\mathbf{L}_{M-1},f_{M-1}),\\
&u_M=u_M+\mathbf{I}_{M-1}^{M}u_{M-1},\\
&u_M=\mathrm{MV}_M(u_M,\mathbf{L}_M,f_M).
\end{aligned}$$

FMV 循环的递归如下:

(1) 如果 Ω_M 为最粗网格,则转到(3),否则

$$\begin{aligned}
&f_{M-1}=\mathbf{I}_M^{M-1}(f_M-A_Mu_M),\\
&u_{M-1}=0,\\
&u_{M-1}=\mathrm{FMV}_{M-1}(u_{M-1},\mathbf{L}_{M-1},f_{M-1}).
\end{aligned}$$

(2) 修正 $u_M = u_M + I_{M-1}^M u_{M-1}$.

(3) $u_M = MV_M(u_M, L_M, f_M)$.

4　二维多重网格诸元素

4.1　差分格式

对二维模型问题(1-3),熟知的五点差分格式是(1-7)式.当 $h_x = h_y = h$ 时,对每点(x_i, y_j)上的函数值 u_{ij} 与周围邻点的关系可表示为

$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} u_k(x,y) = h_k^2 f_k, \quad \forall (x,y) \in \Omega_k. \tag{4-1}$$

对九点差分格式,可用九点星表示:

$$\frac{1}{6}\begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix} u_k(x,y) = \frac{h_k^2}{12}\begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix} f_k(x,y). \tag{4-2}$$

4.2　光滑迭代

(1-17)式中所列的几种迭代方法对二维泊松方程边值问题(1-3)均有效,对奇异摄动等特殊问题需要适当选取光滑过程.下面仅讨论几种经典的迭代.

4.2.1　G-S 迭代

多维G-S迭代与未知量的编号次序有关.对于二维情况,通过Ω_k上网点即$x^i = (x^i, y^j)$的排列次序确定未知量 $u_{k,i} := u_k(x^i)$ 的顺序.

字典顺序:x方向上的点由左到右,y方向上的线由下往上逐点排列,见图4-1.

旋转字典顺序:y方向上的点由下往上,x方向上的线由左往右逐点排列,见图4-2.

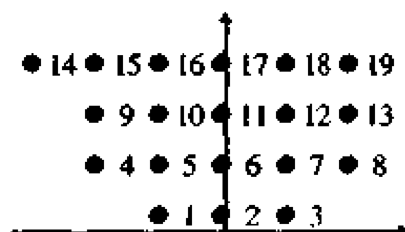


图 4-1

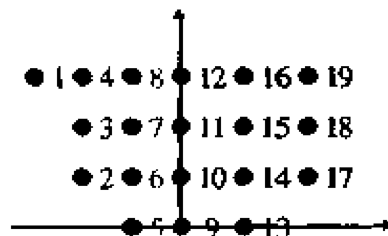


图 4-2

红-黑顺序：设 $\Omega_k=\{(jh_k,lh_k)\mid j,l\in\mathbf{Z}\}$，把 Ω_k 中的点分成两类，称 $j+l$ 为偶数的点为红点，$\Omega_k^r=\{(jh_k,lh_k),j+l\text{ 为偶数}\}$，用■表示，$\Omega_k^b=\Omega_k/\Omega_k^r$ 的点为黑点，用●表示，见图 4-3.

斑马顺序：把 j 为偶数的线称为红线，用●表示，j 为奇数的线称为黑线，用■表示，见图 4-4.

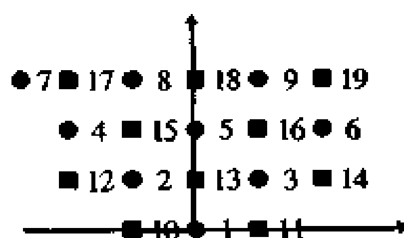


图 4-3

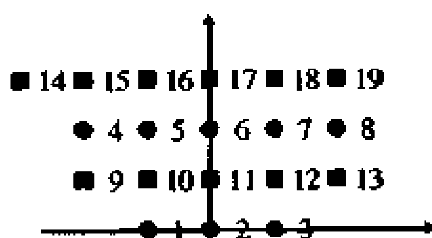


图 4-4

四色顺序：把 Ω_k 分成四个集，

$\Omega_k^1=\{(jh_k,lh_k)\mid j,l\text{ 为偶数}\}$，用●表示.

$\Omega_k^2=\{(jh_k,lh_k)\mid j,l\text{ 为奇数}\}$，用■表示.

$\Omega_k^3=\{(jh_k,lh_k)\mid j\text{ 为偶},l\text{ 为奇数}\}$，用⊗表示.

$\Omega_k^4=\{(jh_k,lh_k)\mid j\text{ 为奇},l\text{ 为偶数}\}$，用×表示.

按字典顺序依次排列 $\Omega_k^1,\Omega_k^2,\Omega_k^3$ 和 Ω_k^4 中的网点，见图 4-5.

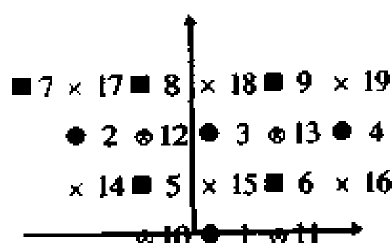


图 4-5

1. 点 G-S 迭代

字典顺序和旋转字典顺序适合于逐点 G-S 迭代，迭代公式为

$$\frac{-u_{i-1,j}^{(n+1)}+2u_{i,j}^{(n+1)}-u_{i+1,j}^{(n)}}{h_x^2}+\frac{-u_{i,j-1}^{(n+1)}+2u_{i,j}^{(n+1)}-u_{i,j+1}^{(n)}}{h_y^2}=f_{i,j},\tag{4-3}$$

其中 $u_{i,j}^{(n+1)}$ 为待求值，$u_{i-1,j}^{(n+1)}$、$u_{i,j-1}^{(n+1)}$ 为已求出的新值，$u_{i+1,j}^{(n)},u_{i,j+1}^{(n)}$ 为上一次迭代之值.

红黑顺序适合于五点差分格式，先依次算红点，再依次算黑点，其迭代公式为

$$\frac{-u_{i-1,j}^{(n)}+2u_{i,j}^{(n+1)}-u_{i+1,j}^{(n)}}{h_x^2}+\frac{-u_{i,j-1}^{(n)}+2u_{i,j}^{(n+1)}-u_{i,j+1}^{(n)}}{h_y^2}=f_{i,j},\tag{4-4}$$

其中红点上的 $u_{i,j}^{(n+1)}$ 为待求量，其相邻的黑点均为前一次的迭代值，黑点上的量为已知量. 下一轮计算时，黑点上的量为待求量，其相邻的红点上均为前一次的迭代值，这时，红点上的量为已知量.

四色顺序适合于九点差分格式. 先依次算 Ω_k^1 中的点，接着依次算 Ω_k^2 中的点，再依次算 Ω_k^3 中的点，最后依次算 Ω_k^4 中的点，这样构成一次迭代.

Ω_k^1 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-\left(u_{i-1,j-1}^{(n)}+u_{i-1,j+1}^{(n)}+u_{i+1,j-1}^{(n)}+u_{i+1,j+1}^{(n)}\right)-4\left(u_{i-1,j}^{(n)}+u_{i+1,j}^{(n)}+u_{i,j-1}^{(n)}+u_{i,j+1}^{(n)}\right)\Big]$$
$$=\frac{1}{12}\left(8f_{i,j}^{(n)}+f_{i-1,j}^{(n)}+f_{i+1,j}^{(n)}+f_{i,j-1}^{(n)}+f_{i,j+1}^{(n)}\right). \tag{4-5a}$$

Ω_k^2 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-\left(u_{i-1,j-1}^{(n+1)}+u_{i-1,j+1}^{(n+1)}+u_{i+1,j-1}^{(n+1)}+u_{i+1,j+1}^{(n+1)}\right)-4\left(u_{i-1,j}^{(n)}+u_{i+1,j}^{(n)}+u_{i,j-1}^{(n)}+u_{i,j+1}^{(n)}\right)\Big]$$
$$=\frac{1}{12}\left(8f_{i,j}^{(n)}+f_{i-1,j}^{(n)}+f_{i+1,j}^{(n)}+f_{i,j-1}^{(n)}+f_{i,j+1}^{(n)}\right). \tag{4-5b}$$

Ω_k^3 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-\left(u_{i-1,j-1}^{(n)}+u_{i-1,j+1}^{(n)}+u_{i+1,j-1}^{(n)}+u_{i+1,j+1}^{(n)}\right)-4\left(u_{i-1,j}^{(n+1)}+u_{i+1,j}^{(n+1)}+u_{i,j-1}^{(n+1)}+u_{i,j+1}^{(n+1)}\right)\Big]$$
$$=\frac{1}{12}\left(8f_{i,j}^{(n)}+f_{i-1,j}^{(n+1)}+f_{i+1,j}^{(n+1)}+f_{i,j-1}^{(n+1)}+f_{i,j+1}^{(n+1)}\right). \tag{4-5c}$$

Ω_k^4 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-\left(u_{i-1,j-1}^{(n+1)}+u_{i-1,j+1}^{(n+1)}+u_{i+1,j-1}^{(n+1)}+u_{i+1,j+1}^{(n+1)}\right)-4\left(u_{i-1,j}^{(n+1)}+u_{i+1,j}^{(n+1)}+u_{i,j-1}^{(n+1)}+u_{i,j+1}^{(n+1)}\right)\Big]$$
$$=\frac{1}{12}\left(8f_{i,j}^{(n)}+f_{i-1,j}^{(n+1)}+f_{i+1,j}^{(n+1)}+f_{i,j-1}^{(n+1)}+f_{i,j+1}^{(n+1)}\right). \tag{4-5d}$$

在以上(4-5)式的每个公式中仅 $u_{i,j}^{(n+1)}$ 为未知量，其余均为已知量，仍属点 G-S 公式. 每个公式中待求点与周围邻点的关系见图 4-6.

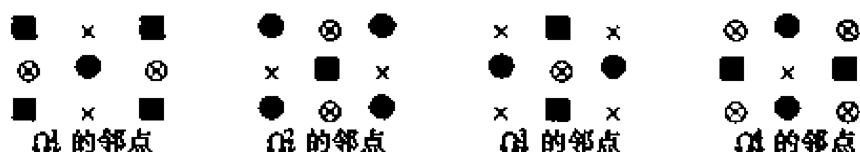


图 4-6

2. 块 G-S 迭代

如果将图 4-1 中的字典顺序网点进行分组，在 Ω_k 上，y_j 为常量的每条线上的网点为一组，共分成四组，分别为 $I_k^1=\{1,2,3\}$，$I_k^2=\{4,5,6,7,8\}$，$I_k^3=\{9,10,11,12,13\}$，$I_k^4=\{14,15,16,17,18,19\}$ 对五点差分格式，类似于(1-8)式，可得矩阵方程组

$$\begin{bmatrix} A_1 & -I & & \\ -I & A_2 & -I & \\ & -I & A_3 & -I \\ & & -I & A_4 \end{bmatrix}\begin{bmatrix} u_k^1 \\ u_k^2 \\ u_k^3 \\ u_k^4 \end{bmatrix}=\begin{bmatrix} f_k^1 \\ f_k^2 \\ f_k^3 \\ f_k^4 \end{bmatrix} \tag{4-6}$$

其中 u_k^1、$f_k^1 \in \mathbf{R}^3$，$A_1 \in \mathbf{R}^{3\times3}$，$u_k^2$、$f_k^2$、$u_k^3$、$f_k^3 \in \mathbf{R}^5$，$A_2$、$A_3 \in \mathbf{R}^{5\times5}$，$u_k^4$、$f_k^4 \in \mathbf{R}^6$，$A_4 \in \mathbf{R}^{6\times6}$ $A_i(i = 1,2,3,4)$ 均为三对角矩阵，从而形成一次求解一条线上未知量的块迭代公式：

$$A_i u_k^{i(n+1)} = a_i u_k^{(i-1)(n+1)} + b_i u_k^{(i+1)(n)} + f_k^i,\qquad i = 1,2,3,4,\tag{4-7}$$

其中 $a_0 = b_4 = 0$，$a_1 = a_2 = a_3 = 1$，$b_1 = b_2 = b_3 = 1$.(4-7)式右端均为已知向量，可直接用追赶法求解此方程组.称以上的块迭代法为字典 x - 线 **G - S** 迭代法.

若用旋转字典顺序，在 Ω_k 上，x_j 为常量的每条线上的网点为一组，则类似地可构成字典 y- 线 **G-S** 迭代法.

按斑马顺序，如图 4-4，构造相应的 G-S 迭代称为 x - 斑马线 **G-S** 迭代法.同理可构造 y - 斑马线 **G-S** 迭代法.

在通常情况下，线 G-S 迭代法的光滑性质稍比点 G-S 迭代法好.

4.2.2　阻尼雅可比方法

对于二维模型(1-3)，其离散方程为(1-7)和(1-8)式.令 $A = h^2 L_h$，当 $h_x = h_y = h = 1/N$ 时，A 的特征值和特征向量分别为

$$\lambda_k(A) = 4\left(1 - \frac{1}{2}(\cos(k_1\pi h) + \cos(k_2\pi h))\right),\quad 1 \leqslant k_1, k_2 \leqslant N - 1,\tag{4-8}$$

$$w_{k,j}(A) = 2\sin(k_1\pi x)\sin(k_2\pi y),\quad 1 \leqslant k_1, k_2 \leqslant N - 1.\tag{4-9}$$

令 $\boldsymbol{x} = (x, y) \in \Omega_h$.

阻尼雅可比或雅可比 -ω 松弛公式是

$$\overline{u}_h^{(n)} = u_h^{(n)} + \omega(z_h(x) - u_h^{(n)}),\tag{4-10}$$

$$\frac{4}{h^2} z_h(\boldsymbol{x}) + L_h u_h^{(n)} - \frac{4}{h^2} u_h^{(n)} = f_h(\boldsymbol{x}),\quad \boldsymbol{x} \in \Omega_h,\tag{4-11}$$

其迭代矩阵为

$$B_{J\omega} = I_h - \frac{\omega}{4} A,\tag{4-12}$$

易知其特征值和特征向量分别为

$$\lambda_k(B_{J\omega}) = 1 - \frac{\omega}{2}\left(2 - \cos\frac{k_1\pi}{N} - \cos\frac{k_2\pi}{N}\right),\quad 1 \leqslant k_1, k_2 \leqslant N - 1,\tag{4-13}$$

$$W_k(B_{J\omega}) = 2\sin(k_1\pi x)\sin(k_2\pi y),\quad 1 \leqslant k_1, k_2 \leqslant N - 1,$$

其中 $\boldsymbol{k} = (k_1, k_2)$.若迭代初始误差按 $B_{J\omega}$ 的特征向量展开，则有

$$e^{(0)} = \sum_{|\boldsymbol{k}| \leqslant N-1} d_k W_k(B_{J\omega}),$$

$$|\boldsymbol{k}| = \max(k_1, k_2)$$

其中 d_k 为常数，经 n 次迭代后

$$e^{(n)} = \sum_{|\boldsymbol{k}| \leqslant N-1} d_k \lambda_k^n(B_{J\omega}) W_k(B_{J\omega}).$$

把级数分成高频和低频两部分

低频： W_k，　　$|k|<N/2$，

高频： W_k，　　$N/2\leqslant|k|\leqslant N-1$，

选 ω 使 $\max\{\lambda_{N/2}(B_{J\omega}),\lambda_N(B_{J\omega})\}$ 达到最小，即

$$\lambda_{N/2}(B_{J\omega})=-\lambda_N(B_{J\omega}),$$

求出使阻尼雅可比迭代法收敛最快的最佳松弛因子 $\omega_{opt}=2/3$. 同一维一样，$\omega_{opt}=2/3$ 的雅可比迭代法能再次很好地阻尼高频分量，这时

$$\lambda_k(B_{J\omega})=1-\frac{4}{3}\sin^2\left(\frac{k_1\pi}{2N}\sin^2\frac{k_2\pi}{2N}\right). \tag{4-14}$$

当 $N/2\leqslant|k|\leqslant N-1$ 时，$|\lambda_k(B_{J2/3})|\leqslant 1/3$，即阻尼雅可比迭代法的光滑因子 $\mu(B_{J2/3})=1/3$，故它对高频分量有很好的光滑作用.

阻尼雅可比迭代法与未知量的排列顺序无关.

对于块阻尼雅可比法，仅有阻尼 x- 线雅可比迭代法和阻尼 y- 线雅可比迭代法，它们均与块的排列顺序无关.

4.3 网格粗化

常用的网格粗化方式有：

标准粗化：$h_{k-1}=2h_k$（见图4-7）. 我们只要求 $h_{k-1}^x/h_k^x=h_{k-1}^y/h_k^y=2$，可以假定 x 和 y 方向上的网格步长 h_k^x 和 h_k^y 是不同的.

半粗化：x 粗化，由 $h_{k-1}^y=h_k^y$ 和 $h_{k-1}^x/h_k^x=2$ 定义粗网格，或 y 粗化，由 $h_{k-1}^x=h_k^x$ 和 $h_{k-1}^y/h_k^y=2$ 定义粗网格（见图4-8）.

红黑粗化：仅考虑方网格，即 $h_k^x=h_k^y$，又称为$\sqrt{2}$ 的均匀网格粗化（见图4-9）.

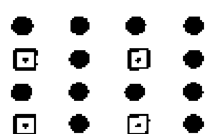


图4-7

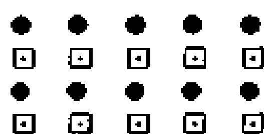


图4-8

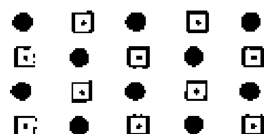


图4-9

4.4 延拓或插值

4.4.1 分片线性插值作为延拓

对二阶或四阶微分方程边值问题，一般用分片线性插值就足够了. 这里只介绍九点延拓和七点延拓.

1. 九点延拓

用如下的九点星模式符号描述，

$$I_{k-1}^k=\frac{1}{4}\left]\begin{matrix}1&2&1\\2&4&2\\1&2&1\end{matrix}\right[,$$

它表示粗网上的函数值按此权系数分配到邻近的细网点上去.取一个标准单元$(0,2h)\times(0,2h)$为例.如果粗细网点重合,则在该点细网函数值就等于粗网点函数值,即

$$\begin{cases} u_k(0,0) = u_{k-1}(0,0), & u_k(0,2h) = u_{k-1}(0,2h), \\ u_k(2h,0) = u_{k-1}(2h,0), & u_k(2h,2h) = u_{k-1}(2h,2h). \end{cases} \tag{4-15}$$

若细网点位于两个粗网点之间,则用这两个粗网点上的函数值作线性插值,就得到细网点上的函数值,

$$\begin{cases} u_k(0,h) = (u_{k-1}(0,0) + u_{k-1}(0,2h))/2, \\ u_k(h,0) = (u_{k-1}(0,0) + u_{k-1}(2h,0))/2, \\ u_k(2h,h) = (u_{k-1}(2h,0) + u_{k-1}(2h,2h))/2, \\ u_k(h,2h) = (u_{k-1}(0,2h) + u_{k-1}(2h,2h))/2, \end{cases} \tag{4-16}$$

有两种插值方法求位于四个粗网点中心的细网点函数值,即

$$u_k(h,h) = (u_{k-1}(0,0) + u_{k-1}(2h,0) + u_{k-1}(0,2h) + u_{k-1}(2h,2h))/4 \tag{4-17}$$

或

$$u_k(h,h) = (u_{k-1}(0,0) + u_{k-1}(2h,2h))/2. \tag{4-18}$$

对于其它网格单元$(2ih_k,2ih_k+2h_k)\times(2jh_k,2jh_k+2h_k)$,可得到类似的双线性插值公式.

九点延拓对应于双线性有限元.对于$0\leqslant x、y\leqslant 2h$,$u_k = I_{k-1}^k u_{k-1}$的$u_k(x,y)$与$u_{k-1}$在$(0,0)$,$(2h,0)$,$(0,2h)$,$(2h,2h)$点上函数值的双线性插值相同.

2.七点延拓

用如下的七点星模式符号描述:

$$I_{k-1}^k = \left]\begin{matrix} 0 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ 1/2 & 1/2 & 0 \end{matrix}\right[,$$

七点延拓对应于三角形上的线性有限元.对于$0\leqslant x、y\leqslant 2h$,$u_k = I_{k-1}^k u_{k-1}$的$u_k(x,y)$与$u_{k-1}$在$(0,0)$,$(0,2h)$,$(2h,2h)$点上函数值的线性插值相同.

分片线性插值推广到三维或更高维是显然的.

4.4.2　高阶插值

对于四阶微分方程边值问题,用分片二次延拓常常是足够的,而用三次插值常常更好些,因为它并不比二次插值公式复杂.

若已知粗网上的插值节点$(x-3h_k,y)$,$(x-h_k,y)$,$(x+h_k,y)$,$(x+3h_k,y)$,则插值点$(x,y)=(ih_k,jh_k)$(i奇数,j偶数)的三次拉格朗日(Lagrange)插值$(I_{k-1}^k u_{k-1})(x,y)$为

$$\begin{aligned} u_k(x,y) = & -\frac{1}{16}[u_{k-1}(x-3h_k,y) + (x+3h_k,y)] + \\ & \frac{9}{16}[u_{k-1}(x-h_k,y) + (u_{k-1}(x+h_k,y))], \\ & \qquad x\in[x-3h_k,x+3h_k]. \end{aligned} \tag{4-19}$$

当点(x,y)中的j为奇数时,我们能重复y方向的插值过程.由插值公式的余项可知,三次拉格朗日插值对于任一双三次函数$\sum_{i=0}^{3}\sum_{j=0}^{3}\alpha_{ij}x^iy^j$是精确的.

如果某个插值点有一个或两个邻点不在域内,则不能用三次插值公式(4-19);如在边界附近,取边界点$(x-sh_k,y)$为插值节点,再取两个内部粗网点$(x+h_k,y)$、$(x+3h_k,y)$为插值节点并且边界条件是齐次狄利克雷条件$u(x-sh_k,y)=0$,那么用如下的二次拉格朗日插值公式去求细网点上的函数值$u_k(x,y)$和$u_k(x+2h_k,y)$更方便:

$$u_k(x,y)=\frac{s}{1+s}\cdot\frac{3}{2}u_{k-1}(x+h_k,y)-\frac{s}{3+s}\cdot\frac{1}{2}u_{k-1}(x+3h_k,y), \tag{4-20}$$

$$u_k(x+2h_k,y)=\frac{2+s}{1+s}\cdot\frac{1}{2}u_{k-1}(x+h_k,y)+\frac{2+s}{3+s}\cdot\frac{1}{2}u_{k-1}(x+3h_k,y). \tag{4-21}$$

在区域内部的单边二次插值公式为

$$u_k(x,y)=-\frac{1}{8}u_{k-1}(x-3h_k,y)+\frac{3}{4}u_{k-1}(x-h_k,y)+\frac{3}{8}u_k(x+h_k,y). \tag{4-22}$$

4.5 限 制

1. 九点限制

用如下的九点星模式符号描述,

$$\mathbf{I}_k^{k-1}=\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}.$$

九点限制即完全加权算子,若有粗网点$(2h,2h)$,按完全加权算子运算,该粗网点上的函数值与周围细网点上函数值的关系如下:

$$\begin{aligned}u_{k-1}(2h,2h)=\frac{1}{16}[&4u_k(2h,2h)+2(u_k(2h,h)+u_k(h,2h)+\\&u_k(3h,2h)+u_k(2h,3h))+u_k(h,h)+u_k(3h,h)+\\&u_k(h,3h)+u_k(3h,3h)],\end{aligned} \tag{4-23}$$

2. 七点限制

用如下的七点星模式符号描述,

$$\mathbf{I}_k^{k-1}=\frac{1}{4}\begin{bmatrix}0&1/2&1/2\\1/2&1&1/2\\1/2&1/2&0\end{bmatrix}.$$

显然,按七点加权运算,粗网点$(2h,2h)$上的函数值与周围细网点上函数值的关系如下:

$$u_{k-1}(2h,2h)=\frac{1}{4}[u_k(2h,2h)+\frac{1}{2}(u_k(2h,h)+u_k(h,2h)+$$

$$u_k(3h,2h) + u_k(2h,3h) + u_k(h,h) + u_k(3h,3h))]. \tag{4-24}$$

对多重网格的每一层 k 定义如下内积：

$$(u_k, v_k) = \sum_{x\in\Omega_M} h_k^d u_k(x)\overline{v_k(x)},$$

其中 d 是维数，$\Omega \in \mathbf{R}^d$. 从而可以定义**伴随映射**

$$(\mathbf{I}_{k-1}^k u, v)_k = (u, (I_{k-1}^k)^* v)_{k-1},$$

称 $(\mathbf{I}_{k-1}^k)^*$ 为 $\mathbf{I}_{k-1}^k$ 的伴随映射.

当 $d = 2$ 时，易知

$$\mathbf{I}_k^{k-1} = (\mathbf{I}_{k-1}^k)^* = \frac{h^2}{(2h)^2}(\bar{\mathbf{I}}_{k-1}^k)^{\mathrm{T}} = \frac{1}{4}(\mathbf{I}_{k-1}^k)^{\mathrm{T}}.$$

因此，九点限制是九点延拓的伴随，七点限制是七点延拓的伴随.

关于延拓和限制的附注：

1° 最显然的一种限制算子是直接映射(injection)，即粗网向量直接取相应细网点上的值，这种算子没有伴随.

2° 令 $2m$ 为所求解的微分方程的阶，m_p 为延拓的阶，m_R 为限制的阶，则它们应满足条件 $m_p + m_R > 2m$.

5 非线性多重网格法

非线性微分方程，如二维方程

$$-\Delta u(x,y) = e^{u(x,y)},$$

其右端项关于 $u(x,y)$ 是非线性的，或边界条件是非线性的. 把微分方程和边界条件结合起来，可以得到抽象方程

$$\hat{\mathrm{L}}u = f. \tag{5-1}$$

前述的线性多重网格方法完全适合于这类非线性边值问题，这里仅讨论两种方法. 5.1 节讨论基于线性化的方法. 5.2 节讨论非线性多重网格方法的本质处理.

5.1 牛顿多重网格法

牛顿多重网格法是一种基于线性化的方法. 令

$$\varphi(u) = \hat{\mathrm{L}}u - f = 0. \tag{5-2}$$

在 $\Omega_k, k = 0,1,2,\cdots,M$, 上分别用五点差分离散(5-2)式后，得到如下离散形式的非线性方程组

$$\varphi_k(u_k) = 0, \quad k = 0,1,2,\cdots,M, \tag{5-3}$$

其中 φ_k、$u_k \in \mathbf{R}^{N_k}$, $N_k = I_k \times J_k$, I_k 为 x 方向的网点数，J_k 为 y 方向的网点数.

将 $\varphi_k(u_k)$ 在 $u_k^{(n)}$ 处作泰勒展开，舍去 $(u_k - u_k^{(n)})$ 的高阶项，留下线性项，这本

质上是把 $\varphi_k(u_k)$ 线性化，得到如下的牛顿迭代公式：

$$\begin{cases}\Psi_k(u_k^{(n)})w_k = d_k^{(n)},\\ u_k^{(n+1)} = u_k^{(n)} + w_k,\end{cases} \quad n = 0,1,2,\cdots, \tag{5-4}$$

其中

$$d_k^{(n)} = -\varphi_k(u_k^{(n)}),$$

$$\Psi_k(u_k) = \frac{\partial \varphi_k}{\partial u_k}(u_k) = \begin{bmatrix} \frac{\partial \varphi_k^1}{\partial u_k^1} & \frac{\partial \varphi_k^1}{\partial u_k^2} & \cdots & \frac{\partial \varphi_k^1}{\partial u_k^n} \\ \frac{\partial \varphi_k^2}{\partial u_k^1} & \frac{\partial \varphi_k^2}{\partial u_k^2} & \cdots & \frac{\partial \varphi_k^2}{\partial u_k^n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \varphi_k^n}{\partial u_k^1} & \frac{\partial \varphi_k^n}{\partial u_k^2} & \cdots & \frac{\partial \varphi_k^n}{\partial u_k^n} \end{bmatrix}(u_k) \tag{5-5}$$

这是向量 φ_k 对向量 u_k 的导数，即在 u_k 处的雅可比矩阵.

牛顿多重网格算法如下：

(1) 给定初值 $u_k^{(0)}$.

(2) 进行迭代，$n = 0,1,2,\cdots$

1) 计算 $d_k^{(n)} = -\varphi_k(u_k^{(n)}), \Psi_k^{(n)} = \frac{\partial \varphi_k}{\partial u_k}(u_k^{(n)})$,

2) 以 $w_k^{(0)}$ 为初值，对 $\Psi_k^{(n)}W_k = d_k^{(n)}$ 执行一个线性多重网格迭代步，得到 $w_k^{(1)}$.

3) 计算 $u_k^{(n+1)} = u_k^{(n)} + w_k^{(1)}$.

这是一个典型的牛顿多重网格迭代算法.

5.2　非线性多重网格法

利用非线性多重网格算法能避免牛顿迭代中的线性化，这个算法与线性多重网格法一样有效. 下面讨论这个算法，注意它与线性情况的不同之处.

5.2.1　非线性雅可比-ω 迭代方法

在多重网格法中，我们感兴趣的是非线性迭代法对高频误差分量的光滑性质. 众所周知，在迭代方法中引入松弛参数 ω，当 $0 < \omega < 1$ 时，低松弛校正可以扩大收敛域，如将牛顿迭代法(5-4)的第二式改为 $u_k^{(n+1)} = u_k^{(n)} + \omega w_k$. 本节着重讨论雅可比-牛顿 ω 松弛法和雅可比-皮卡(Picard) ω 松弛法.

考虑非线性问题(5-1)的离散方程组

$$\hat{L}_h u_h = f_h, \tag{5-6}$$

为讨论方便起见，将该非线性差分方程写成如下形式：

$$\hat{L}_h u_h = L_h u_h + g(x, u_h) = f_h \tag{5-7}$$

其中 L_h 为线性差分算子. 为讨论方便起见，设 L_h 为五点差分算子，$g(x, u_h)$ 为 u_h

的线性或非线性函数.

类似于(4-10)、(4-11)式,一个非线性雅可比 ω 松弛定义为

$$\bar{u}_h^{(n)} = u_h^{(n)} + \omega(z_h - u_h^{(n)}), \tag{5-8}$$

$$\frac{4}{h^2}z_h + (\mathbf{L}_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)}) + g(x, z_h) = f_h. \tag{5-9}$$

将(5-9)式中的非线性项 $g(x,z_h)$ 在 $u_h^{(n)}$ 处作泰勒展开,仅留下关于 $(z_h - u_h^{(n)})$ 线性项,则被线性化为

$$\begin{aligned}\frac{4}{h^2}z_h + (\mathbf{L}_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)}) + g(x, u_h^{(n)})\\ + \frac{\partial g}{\partial u}(x, u_h^{(n)})(z_h - u_h^{(n)}) = f_h,\end{aligned} \tag{5-10}$$

称(5-8)和(5-10)式为**雅可比-牛顿-ω 松弛法**.

若用 $g(x,u_h^{(n)})$ 代替方程(5-9)中的 $g(x,z_h)$,则有

$$\frac{4}{h^2}z_h + (\mathbf{L}_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)}) + g(x, u_h^{(n)}) = f_h, \tag{5-11}$$

称(5-8)式和(5-11)式为**雅可比-皮卡(Picard)-ω 松弛法**.

为具体讨论以上两种迭代方法的光滑效应,假设 g 为 u 的线性函数,即

$$g(x,u) = cu, \qquad c > 0. \tag{5-12}$$

显然 $\frac{\partial g}{\partial u} = c$,将它们代入(5-10)和(5-11)式,则雅可比-牛顿 ω 松弛法的迭代矩阵为

$$\boldsymbol{B}_{\mathrm{JN}\omega} = (1 - \frac{\omega_N ch^2}{4 + ch^2})\mathbf{I}_h - \frac{\omega_N h^2}{4 + ch^2}\mathbf{L}_h; \tag{5-13}$$

雅可比-皮卡 ω 松弛法的迭代矩阵为

$$\boldsymbol{B}_{\mathrm{JP}\omega} = (1 - \frac{\omega_P ch^2}{4})\mathbf{I}_h - \frac{\omega_P h^2}{4}\mathbf{L}_h. \tag{5-14}$$

当 $\omega_N = \frac{4 + ch^2}{4}\omega_P$ 时,$\boldsymbol{B}_{\mathrm{JN}\omega} = \boldsymbol{B}_{\mathrm{JP}\omega}$. 由(4-12)和(4-13)式可知

$$\boldsymbol{B}_{\mathrm{J}\omega} = \mathbf{I}_h - \frac{\omega h^2}{4}\mathbf{L}_h, \tag{5-15}$$

$$\lambda(\boldsymbol{B}_{\mathrm{J}\omega}) = 1 - \frac{\omega}{2}\left(2 - \cos\frac{k_1\pi}{N} - \cos\frac{k_2\pi}{N}\right), \quad 1 \leqslant k_1, k_2 \leqslant N - 1. \tag{5-16}$$

将(5-15)和(5-16)式代入(5-13)和(5-14)式,得

$$\boldsymbol{B}_{\mathrm{JN}\omega} = \frac{4}{4 + ch^2}\boldsymbol{B}_{\mathrm{J}\omega} + \frac{(1 - \omega_N)ch^2}{4 + ch^2}, \quad \boldsymbol{B}_{\mathrm{JP}\omega} = \boldsymbol{B}_{\mathrm{J}\omega} - \frac{\omega_P ch^2}{4},$$

易得它们的特征值分别为

$$\lambda(\boldsymbol{B}_{\mathrm{JN}\omega}) = \frac{4}{4 + ch^2}\lambda(\boldsymbol{B}_{\mathrm{J}\omega}) + \frac{(1 - \omega_N)ch^2}{4 + ch^2},$$

$$\lambda(\boldsymbol{B}_{\mathrm{JP}\omega}) = \lambda(\boldsymbol{B}_{\mathrm{J}\omega}) - \frac{\omega_P ch^2}{4},$$

雅可比-牛顿 ω 松弛法的光滑因子为

$$\mu(\boldsymbol{B}_{\mathrm{JN}\omega}) = \max_{N/2 \leq |k| \leq N-1} |\lambda(\boldsymbol{B}_{\mathrm{JN}}\omega)|$$
$$= \max\left\{ \left| 1 - \omega_N\left(1 - \frac{2\cos(\pi h)}{4 + ch^2}\right) \right|, \left| 1 - \omega_N\left(1 + \frac{4\cos(\pi h)}{4 + ch^2}\right) \right| \right\},$$
(5-17)

其中 $k = (k_1, k_2)$，$|k| = \max\{k_1, k_2\}$. 上式括弧内的两个值，它们是 k_1 和 k_2 分别取 $N/2$ 和 $N-1$ 时得到的 $\lambda(\boldsymbol{B}_{\mathrm{JN}\omega})$ 以及 k_1 和 k_2 均取 $N-1$ 时得到的 $\lambda(\boldsymbol{B}_{\mathrm{JN}\omega})$. 从上式可知对固定的 $\omega_N(0 < \omega_N < 1)$ 和固定的 h，当 $c \to \infty$ 时，有

$$\lim_{c \to \infty} \mu(\boldsymbol{B}_{\mathrm{JN}\omega}) = |1 - \omega_N|.$$

可以求出雅可比 - 牛顿 ω 松弛法的最佳松弛因子和最佳光滑因子分别为

$$\omega_{\mathrm{opt}} = \frac{4 + ch^2}{4 + ch^2 + \cos(\pi h)}, \quad \mu_{\mathrm{opt}} = \frac{3\cos(\pi h)}{4 + ch^2 + \cos(\pi h)}.$$

易见对固定的 h，雅可比 - 牛顿 ω 松弛法的光滑因子随 c 的增加而变好.

对雅可比 - 皮卡 ω 松弛法，将 $\omega_N = \dfrac{4 + ch^2}{4}\omega_P$ 代入(5-17)式，则有雅可比 - 皮卡 ω 松弛法的光滑因子

$$\mu(\boldsymbol{B}_{\mathrm{JP}\omega}) = \max\left\{ \left| 1 - \omega_P \frac{4 + ch^2}{4}\left(1 - \frac{2\cos(\pi h)}{4 + ch^2}\right) \right|, \left| 1 - \omega_P \frac{4 + ch^2}{4}\left(1 + \frac{4\cos(\pi h)}{4 + ch^2}\right) \right| \right\}$$
$$= \max\left\{ \left| 1 - \omega_P\left(1 + \frac{ch^2 - 2\cos(\pi h)}{4}\right) \right|, \left| 1 - \omega_P \frac{4 + ch^2 + 4\cos(\pi h)}{4} \right| \right\}.$$

易知当固定 $\omega_P(0 < \omega_P < 1)$ 和 h，随 c 增加到充分大时，雅可比 - 皮卡 ω 松弛法没有光滑性质.

5.2.2　亏损方程

考虑非线性问题(5-1)的离散方程组(5-6)

$$\widehat{\mathrm{L}}_h u_h = f_h,$$

其中 $\widehat{\mathrm{L}}_h$ 为差分算子，h 为网格步长. 令 u_h^* 为精确解，u_h 为近似解，v_h 为误差. 将 $u_h^* = u_h + v_h$ 代入上式，有

$$\widehat{\mathrm{L}}_h(u_h + v_h) = f_h.$$

当 $\widehat{\mathrm{L}}_h$ 为线性算子 L_h 时，则

$$\mathrm{L}_h u_h + \mathrm{L}_h v_h = f_h,$$

亏损方程为

$$\mathrm{L}_h v_h = d_h,$$

其中 $d_h = f_h - \mathrm{L}_h u_h$. 从亏损方程求出 v_h，用它对 u_h 进行修正.

当 $\widehat{\mathrm{L}}_h$ 为非线性算子时，$\widehat{\mathrm{L}}_h$ 不能分别作用于 u_h 和 v_h，则

$$\widehat{\mathrm{L}}_h(u_h + v_h) - \widehat{\mathrm{L}}_h u_h = f_h - \widehat{\mathrm{L}}_h u_h.$$

仍令 $d_h = f_h - \widehat{\mathrm{L}}_h u_h$，则亏损方程为

$$\widehat{\mathrm{L}}_h(u_h + v_h) - \widehat{\mathrm{L}}_h u_h = d_h, \tag{5-18}$$

在粗网 Ω_H 上的亏损方程为

$$\widehat{\mathrm{L}}_H(I_h^H u_h + v_H) - \widehat{\mathrm{L}}_H(I_h^H u_h) = \mathrm{I}_h^H d_h, \tag{5-19}$$

其中 I_h^H 为限制算子.与线性情况不同,不是在粗网上直接求解校正量 v_H,而是求解完全逼近量 $u_H = I_h^H u_h + v_H$,即求解方程

$$\widehat{\mathrm{L}}_H u_H = d_H + \widehat{L}_H(\mathrm{I}_h^H u_h), \tag{5-20}$$

然后,有校正量

$$v_H = u_H - \mathrm{I}_h^H u_h.$$

5.2.3 FAS 双网格方法

FAS(full approximation storage) 方法是非线性多重网格法,它的特点是不存储校正量 v_H,而是存储完全近似量 $u_H = \mathrm{I}_h^H u_h + v_H$.其中用 G^ν 表示非线性松弛法松弛 ν 次的松弛算子,而不同于线性松弛法中的松弛算子符号 S^ν.

FAS 双网格算法为:

(1) 前光滑:对 $u_h^{(n)}$ 在 Ω_h 上作 ν_1 次非线性松弛,得到

$$\overline{u}_h^{(n)} \xlongequal{\text{def}} \mathrm{G}^{\nu_1}(u_h^{(n)}, \widehat{L}_h, f_h).$$

(2) 粗网修正:

1) 限制近似解: $\overline{u}_H^{(n)} \xlongequal{\text{def}} \mathrm{I}_h^H \overline{u}_h^{(n)}$,

2) 计算网格亏损量:$d_h^{(n)} = f_h - \mathrm{L}_h u_h^{(n)}$,

3) 限制亏损量:$d_H^{(n)} = \mathrm{I}_h^H d_h^{(n)}$,

4) 计算 $f_H = d_H + \widehat{\mathrm{L}}_H \overline{u}_H^{(n)}$,在 Ω_H 上求解亏损方程 $\widehat{\mathrm{L}}_H u_H = f_H$,解得 $\overline{u}_H$,

5) 计算 Ω_H 上的修正量:$v_H^{(n)} = \overline{u}_H - \mathrm{I}_h^H u_h^{(n)}$,

6) 插值修正量:$v_h^{(n)} = \mathrm{I}_H^h v_H^{(n)}$,

7) 对 $\overline{u}_h^{(n)}$ 作修正:$\widehat{u}_h^{(n)} = \overline{u}_h^{(n)} + v_h^{(n)}$.

(3) 后光滑:对 $\widehat{u}_h^{(n)}$ 在 Ω_h 上作 ν_2 次非线性松弛,得到

$$u_h^{(n+1)} := \mathrm{G}^{\nu_2}(\widehat{u}_h^{(n)}, \widehat{\mathrm{L}}_h, f_h).$$

5.2.4 FAS 多重网格法

下面讨论用 FAS 多重网格法求解 Ω_M 上离散的非线性方程组

$$\widehat{L}_M u_M = f_M$$

的计算过程.已知 $u_M^{(n)}$,求 $u_M^{(n+1)} := \mathrm{MFAS}(u_M^{(n)}, \widehat{\mathrm{L}}_M, f_M)$.仿 FAS 双网格法,计算分三个阶段.

(1) 前光滑:对 $u_M^{(n)}$ 在 Ω_M 上作 ν_1 次非线性松弛,得到

$$\overline{u}_M^{(n)} := G^{\nu_1}(u_M^{(n)}, \widehat{\mathbf{L}}_M, f_M).$$

(2) 粗网修正：

1) 限制近似解：$\overline{u}_{M-1}^{(n)} := \mathbf{I}_M^{M-1}\overline{u}_M^{(n)}$,

2) 计算 Ω_M 上的亏损量：$d_M^{(n)} = f_M - \mathbf{L}_M u_M^{(n)}$,

3) 限制亏损量：$d_{M-1}^{(n)} = \mathbf{I}_M^{M-1} d_M^{(n)}$,

4) 计算 $f_{M-1} = d_{M-1} + \widehat{\mathbf{L}}_{M-1}\overline{u}_{M-1}^{(n)}$，在 Ω_{M-1} 上求解非线性亏损方程 $\widehat{\mathbf{L}}_{M-1}u_{M-1} = f_{M-1}$，即计算 $\overline{u}_{M-1} := \mathrm{MFAS}(\overline{u}_{M-1}^{(n)}, \widehat{\mathbf{L}}_{M-1}, f_{M-1})$，以 $\overline{u}_{M-1}^{(n)}$ 作为初始值，作 MFAS 网格法的 $r(r \geqslant 1)$ 型迭代，需用到网格 $\Omega_{M-1}, \Omega_{M-2}, \cdots, \Omega_0$,

5) 计算 Ω_{M-1} 上的修正量：$v_{M-1}^{(n)} = \overline{u}_{M-1} - \mathbf{I}_M^{M-1}\overline{u}_M^{(n)}$,

6) 插值修正量：$v_M^{(n)} = \mathbf{I}_{M-1}^M v_{M-1}^{(n)}$,

7) 对 $\overline{u}_M^{(n)}$ 作修正：$\widehat{u}_M^{(n)} = \overline{u}_M^{(n)} + v_M^{(n)}$.

(3) 后光滑：对 $\widehat{u}_M^{(n)}$ 在 Ω_M 上作 ν_2 次非线性松弛，得到

$$u_M^{(n+1)} := G^{\nu_2}(\widehat{u}_M^{(n)}, \widehat{\mathbf{L}}_M, f_M).$$

6　计算机上的执行性能

6.1　数据结构

多重网格法的程序可以做到高度模块化，如松弛、插值和限制均可编成独立的子程序，容易调用，容易替换. 除此之外，对于规则区域还需采用合理的数据结构，当然，对于不规则区域或网格局部加密情况除外.

对于规则区域，只需用两个数组. 一个数组 v 存放每层网格上的解向量和误差向量，从最粗网格到最细网格，逐点依次排列；另一个数组 f 存放每层网格上的右端项和余项，也是从最粗网格到最细网格依次排列. 这两个数组均应包括边界点，对一维问题，第 k 层有 $2^k + 1$ 个点.

图 6-1 描述了间隔数 $N = 16$，网格层数 $M = 4$，V 循环的一维问题的典型数据结构. 说明数据结构随 V 循环在变化.

先初始化每层网格的解向量，即置数组 v 为零，最细网格上的右边数组 f 是已知的，其余层的 f 也置为零. 当 V 循环往粗网方向下降时，每层网格上的松弛把该层解向量数组填满，同时把余项转移到下一层粗网，把该层的右端数组填满. 当 V 循环往细网方向上升时，右边数组不变. 但在每层网格上又进行松弛，重写该层的解数组段，同时把前一层粗网上的解数组段冲成零，目的是为执行另一个 V 循环提供零初始值.

二维问题的数据结构和一维类似，仅仅每层网格上的数据段比一维情况长些.

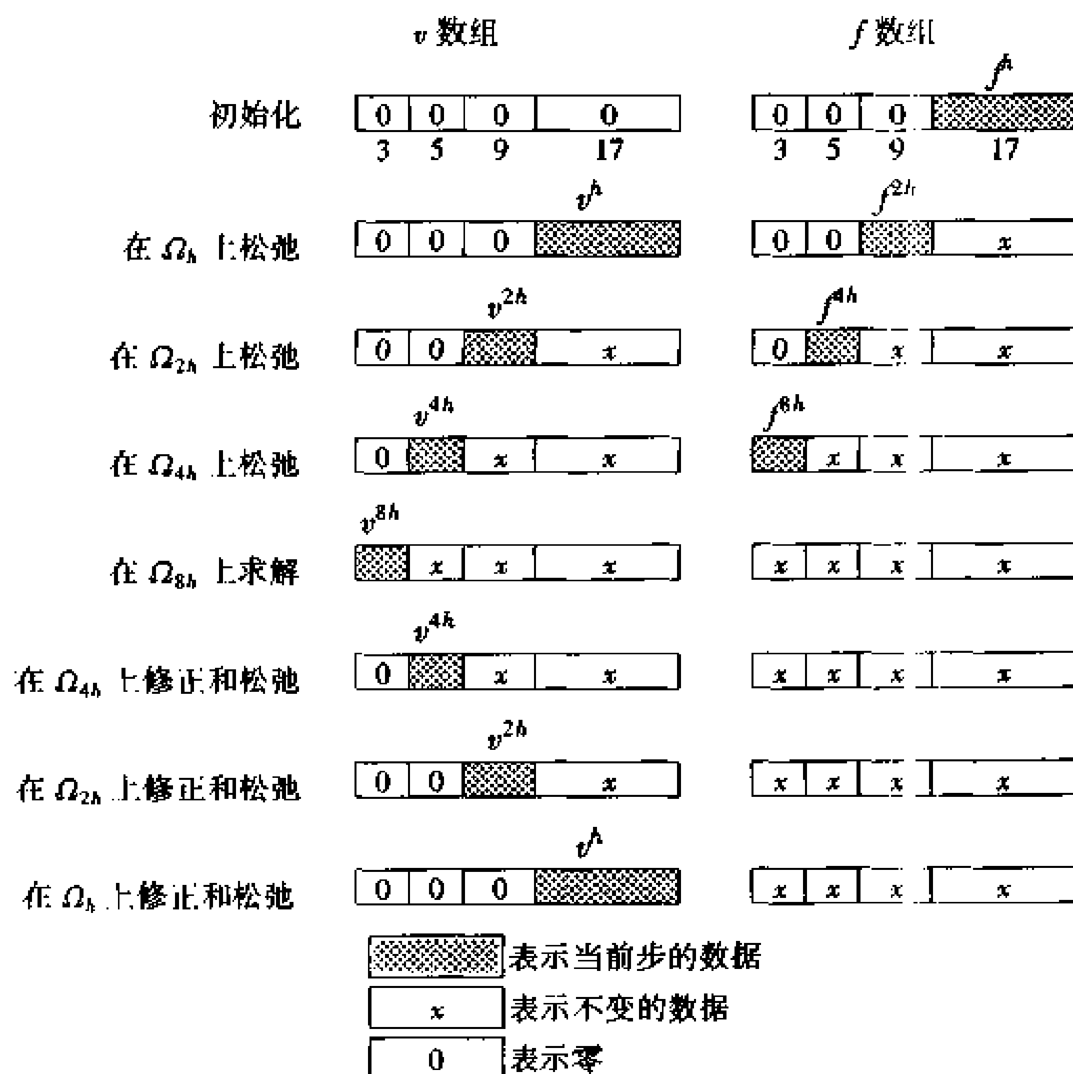


图 6-1

6.2 存 储 量

考虑 d 维 n 层网格，每层网点为 N^d，$N=2^n$. 每层存储 v 和 f 二个数组. 存储量

$$M_d = 2N^d + 2(N/2)^d + \cdots + 2(N/2^n)^d$$
$$= 2N^d(1+2^{-d}+2^{-2d}+\cdots+2^{-nd}) < 2N^d/(1-2^{-d}).$$

当 $d=1$ 时，$M_1 < 2(2N)$，不足 2 倍细网存储量.

当 $d=2$ 时，$M_2 < \frac{4}{3}(2N^2)$ 不足 4/3 倍细网存储量.

当 $d=3$ 时，$M_3 < \frac{8}{7}(2N^3)$ 不足 8/7 倍细网存储量.

因此，多重网格法的存储量随问题维数的增加而相对地下降.

6.3 计 算 量

记 WU(work unit) 为工作单位,即一个 WU 为细网上执行一次松弛所需的算术运算量. $v_1 = v_2 = 1$ 的一个 V 循环多重网格法的工作量为

$$W_{vd} = 2\mathrm{WU}(1 + 2^{-d} + 2^{-2d} + \cdots + 2^{-nd}) < \frac{2}{1 - 2^{-d}}\mathrm{WU},$$

则有

$$W_{v_1} < 4\mathrm{WU}, \quad W_{v_2} < \frac{8}{3}\mathrm{WU}, \quad W_{v_3} < \frac{16}{7}\mathrm{WU}.$$

$v_1 = v_2 = 1$ 的一个 FMV 多重网格法的工作量为

$$W_{\mathrm{FMV}d} = \frac{2\mathrm{WU}}{1 - 2^{-d}}(1 + 2^{-d} + 2^{-2d} + \cdots + 2^{-nd}) < \frac{2}{(1 - 2^{-d})^2}\mathrm{WU},$$

则有

$$W_{\mathrm{FMV1}} < 8\mathrm{WU}, \quad W_{\mathrm{FMV2}} < \frac{7}{2}\mathrm{WU}, \quad W_{\mathrm{FMV3}} < \frac{5}{2}\mathrm{WU}.$$

设 p 为微分方程的阶, M 是一个常数, ϵ 是一个小量为解的计算精度,则在 $h < (\epsilon/(2M))^{1/p}$ 条件下(见[5]),多重网格法收敛到截断误差级所需的运算量为

V 循环多重网格法的总运算量为 $O(N^d \log N)$,

FMV 多重网格法的总运算量为 $O(N^d)$,

这比求解线性代数方程组的直接方法的计算速度还要快.

6.4 数值实例

为了说明多重网格方法的计算效率,用以下的泊松方程边值问题为例:

$$\begin{cases} -\Delta u = 2, \quad 0 < x < 1, 0 < y < 1, \\ u(0,y) = y(1-y), \\ u(1,y) = y(1-y), \\ u(x,0) = x(1-x), \\ u(x,1) = x(1-x). \end{cases}$$

当 $h = 1/256$ 时,用三种迭代法,即逐次超松弛法(SOR法),交替方向隐式方法(ADI法)和 FMG 多重网格法在 IBM/370-158 上进行计算,计算结果列于表 6-1.

表 6-1

方 法	运算次数	计算时间 CPU/s	$\dfrac{\lVert u_h - \tilde{u}_h \rVert}{\lVert u_h \rVert}$
SOR	$\sim N^{1.5}$	~ 1200	
ADI	$\sim N\log N$	127	0.5×10^{-2}
FMG	$\sim N$	7.7	0.2×10^{-4}

参 考 文 献

1 Fedorenko. The Speed of conyergence of one iterative process. USSR Comp Math and MathPhys, 1964(3): 227 ~ 238

2 Brandt A. Multi-level adaptive technique of fast numerical solution to boundary value problems. Proceeding of Third International Conference on Numerical Methods in Fluid Mechanics. Paris, 1972.

3 Brandt A. Multi-level adaptive solution to boundary-value problem. Mathematics of Computation, 1977, 31(138): 333 ~ 390

4 Hackbusch W A. Multigrid method applied to a boundary value problem with variable cofficients in a rectangle. Report 1977(17).

5 Briggs W L A. Multigrid tutorial. Philadephia: SIAM, 1987.

6 Brandt A. Multigrid techniques: 1984 guide with Application to fluid dynamics. GMD Studien no85, Gesellschaft für Mathematik und Datenverarbeitung. Sankt Augustin, Germany.

7 Hackbusch W. Multigrid method and aplication. Berlin: Springer, 1985.

8 W.哈克布思 著.多重网格方法.北京:科学出版社.1988.

9 McCormick S F. Multigrid method (frontiers in applied mathematics3). Philadelphia: SIAM, 1987.

10 McCormick S F. Multigrid method: theory, aplication and supercomputing. New York, 1988.

11 McCormick S F. Multigrid adaptive methods for partial differential equations (frontiers in applied mathematics 6). Philadelphia: SIAM, 1989.

12 曹志浩 编著.多格子方法.上海:复旦大学出版社.1987.

13 Wesseling P. An introduction to multigrid methods. New York: John Wiley & Sons, 1992.

·计算机数学卷·

第6篇

区域分解方法

编　者　胡齐芽
审校者　罗　平

目　录

引　言

区域分解方法是求解大规模科学技术和工程问题的一种新型算法.随着并行计算机的飞速发展,它已受到计算数学专家和工程技术人员的高度重视.与传统的计算方法相比,区域分解方法的优点集中表现在:

(1) 把大问题化成小问题,缩小计算规模.

(2) 把不规则区域上的问题化成规则区域上的问题求解.

(3) 把复杂模型化为简单模型计算.

(4) 算法是高度并行的,即计算的主要步骤可在各子域上独立进行.

尽管区域分解方法的提出距今仅有20年,它现在仍处在研究、发展阶段(每年召开一次国际学术会议),但这种方法已被应用于石油勘探、飞机设计等许多实际问题的数值模拟中,并且其应用范围在迅速扩大.因此,切实掌握这种计算方法的原理对即将跨入21世纪的科技工作者是非常必要的.

本篇对区域分解方法的基本思想、实用算法和主要结果作些介绍.为简化符号,这里只考虑二维情形.由于国内外介绍区域分解方法的著作还很少,所以本篇有些材料取自发表的论文.

1　预备知识

本章给出后面几章中需用到的基本概念和不等式.

1.1　与迭代法有关的概念

考虑线性代数方程组

$$Ax = b, \tag{1-1}$$

其中 A 是 n 阶方阵,b 是 n 维列向量.假定 A 是对称、正定的.

当 A 的阶数 n 很大时,直接求解方程(1-1)非常困难,而代之以迭代方法求解.最简单的一种迭代方法是里查德森(Richardson)迭代.

算法1　里查德森迭代

给定初值 x_0,对 $i=0,1,\cdots$ 执行

$$x_{i+1} = x_i + \tau(b - Ax_i), \tag{1-2}$$

其中实参数 $\tau > 0$ 称为松弛因子.

以 λ_1 和 λ_n 分别表矩阵 A 的最小特征值和最大特征值.以 $\|\cdot\|$ 表向量的欧氏范数.下列结果是基本的:

命题1　当 $0 < \tau < 2/\lambda_n$ 时,算法1-1收敛,且

$$\|x_i - x\| \leqslant \rho \|x_0 - x\|,$$

其中 $\rho = \max\{|1 - \tau\lambda_1|, |\tau\lambda_n - 1|\} < 1$. 特别,当 $\tau = 2/(\lambda_1 + \lambda_n)$ 时,收敛得最快.此时,$\rho_0 = (\lambda_n - \lambda_1)/(\lambda_n + \lambda_1)$ 达到最小.

定义 1 称 λ_n/λ_1 为矩阵 A 的**条件数**,记作 $\kappa(A)$ 或 $\mathrm{cond}(A)$.

该定义只适合于对称、正定矩阵,对一般可逆矩阵,条件数的定义可参见文献[20].

易知,当 $\kappa(A)$ 较小时,最优的 ρ_0 也较小;反之,当 $\kappa(A)$ 很大时,必有 $\rho_0 \to 1$,即算法 1-1 收敛得很慢.通常的解决办法是:构造一适当的 n 阶对称、正定矩阵 B,用里查德森迭代求与(1-1)式同解的方程

$$B^{-1}Ax = B^{-1}b. \tag{1-3}$$

即将(1-2)式代之下列格式:

$$x_{i+1} = x_i + \tau B^{-1}(b - Ax_i). \tag{1-4}$$

B 称为 A 的**预条件子**.

定义 2 若 B 满足下列两个条件:

1° $\kappa(B^{-1}A) \ll \kappa(A)$;

2° B^{-1} 远比 A^{-1} 容易求得,

则称预条件子 B 是有效的.此时,B^{-1} 可认为是 A 的近似逆.

若构造了 A 的有效预条件子 B,则对适当选取的 τ,迭代(1-4)式收敛得较快.但一般难以精确估计矩阵 $B^{-1}A$ 的最大、最小特征值,故这样的 τ 未必能取到.

以下介绍**共轭梯度法**(简称 CG 法).它不仅不需要矩阵特征值的估计,而且是收敛得最快的迭代法.

算法 2 共轭梯度法

步 1 取初始向量 x_0,计算 $r_0 = b - Ax_0$ 和 $p_0 = B^{-1}r_0$.

步 2 对 $k = 1, 2, \cdots$,执行

$$\tau_k = \langle p_{k-1}, p_{k-1}\rangle / \langle Ar_{k-1}, r_{k-1}\rangle,$$

$$x_k = x_{k-1} + \tau_k r_{k-1}, \quad p_k = p_{k-1} + \tau_k A r_{k-1},$$

$$\beta_k = \langle p_k, p_k\rangle / \langle p_{k-1}, p_{k-1}\rangle, \quad r_k = -p_k + \beta_k r_{k-1},$$

其中 $\langle\cdot,\cdot\rangle$ 表向量的欧氏内积.

该算法的收敛性结果也是熟知的(参见文献[20]).

命题 2 对共轭梯度法(算法 2),有

$$\|x_k - x\|_A \leqslant 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|x_0 - x\|_A.$$

其中 $\|\cdot\|_A = \langle A\cdot, \cdot\rangle$ 为 A 生成的范数.

由此可知,CG 法的收敛速度同样依赖于矩阵 A 的条件数 $\kappa(A)$.当 $\kappa(A)$ 很大时,收敛得很慢,只能用 CG 法求解同解的方程(1-3),这种方法称为**预条件共轭梯度法**,简称 **PCG 法**.自然,PCG 法是否有效也取决于预条件子 B 的有效性.

估计 $B^{-1}A$ 的条件数,通常采用内积 $\langle A\cdot, \cdot\rangle$.

命题 3 若存在数 α、$\beta > 0$,使对任意 n 维向量 x 有

$$\alpha(x, Ax) \leqslant (B^{-1}Ax, Ax) \leqslant \beta(x, Ax), \tag{1-5}$$

则 $\kappa(B^{-1}A) \leqslant \beta/\alpha$.

矩阵 A 通常来源于某一离散问题,因此它与离散参数有关,下列概念将经常用到.

定义 3 设 A_1、A_2 是对应于同一离散问题的两个 n 阶对称正定矩阵.若存在与离散参数无关的常数 c 和 C,使对任意 n 维向量 x 有

$$c\langle A_2x, x\rangle \leqslant \langle A_1x, x\rangle \leqslant C\langle A_2x, x\rangle, \tag{1-6}$$

则称 A_1 与 A_2 **谱等价**,记作 $A_1 \sim A_2$.

设 V 是某一有限维线性空间,其内积为$(\cdot,\cdot)$.设 $A: V \to V$ 是对称、正定算子,$a(\cdot,\cdot): V \times V \to \mathbb{R}$ 是由 A 诱导的双线性型,即 $a(\cdot,\cdot) = (A\cdot,\cdot)$.以 $\{\varphi_i\}_{i=1}^n$ 表示 V 的一组基函数,记 $a_{ij} = (A\varphi_i, \varphi_j) = a(\varphi_i, \varphi_j)$.

定义 4 将以 a_{ij} 为元素的矩阵称为 A(或 $a(\cdot,\cdot)$)的**刚度矩阵**.

以上所有算法、概念和结果都可换成算子形式.命题 3 还常以双线性型的形式出现,此时(1-5)式应换成

$$\alpha a(v, v) \leqslant b(v, v) \leqslant \beta a(v, v), \quad \forall v \in V.$$

1.2 索伯列夫空间及其性质

设 Ω 是一有界平面区域,$\partial\Omega$ 是其分片光滑的边界.对实数 $p \in [1, +\infty)$ 和非负整数 k,空间 $L^p(\Omega)$、$H^k(\Omega)$ 和 $C_0^\infty(\Omega)$ 按通常的方式定义(参见文献[20]).$H_0^1(\Omega)$ 定义为 $C_0^\infty(\Omega)$ 中的函数在 $H^1(\Omega)$ 度量下的闭包.

对 $\Sigma \subset \partial\Omega$,定义空间

$$H^{\frac{1}{2}}(\Sigma) = \{\varphi \in L^2(\Sigma) \mid |\varphi|_{\frac{1}{2},\Sigma} < +\infty\},$$

其中

$$|\varphi|_{\frac{1}{2},\Sigma}^2 = \int_\Sigma\int_\Sigma \frac{(\varphi(x) - \varphi(y))^2}{|x - y|^2} ds(x) ds(y).$$

定义范数

$$\|\varphi\|_{\frac{1}{2},\Sigma} = (|\varphi|_{\frac{1}{2},\Sigma}^2 + d^{-1}\|\varphi\|_{0,\Sigma}^2)^{\frac{1}{2}}, \tag{1-7}$$

其中 $\|\cdot\|_{0,\Sigma}$ 表 Σ 上的 L^2 范数,d 表 Ω 的"尺寸"(意义见下一章).这样定义范数是为了使后面出现的常数 C 与 d 无关.

以 $H^{-\frac{1}{2}}(\Sigma)$ 表 $H^{\frac{1}{2}}(\Sigma)$ 的对偶空间,其范数定义为

$$|v|_{-\frac{1}{2},\Sigma} = \sup_{\varphi \in H^{\frac{1}{2}}(\Sigma)} \frac{\left|\int_\Sigma v\varphi ds\right|}{\|\varphi\|_{\frac{1}{2},\Sigma}}. \tag{1-8}$$

对 Ω 的边 E,定义空间 $H_{00}^{\frac{1}{2}}(E)$ 如下:以 $\tilde{\varphi}$ 表 φ 的零扩张,令

$$H_{00}^{\frac{1}{2}}(E) = \{\varphi \in L^2(\partial\Omega) \mid \operatorname{supp}\varphi \subset E, \tilde{\varphi} \in H^{\frac{1}{2}}(\partial\Omega)\}.$$

定义相应的范数为

$$\|\varphi\|_{H_{00}^{\frac{1}{2}}(E)} = \left(|\varphi|^2_{H_{00}^{\frac{1}{2}}(E)} + d^{-1}\|\varphi\|^2_{0,E}\right)^{\frac{1}{2}}, \tag{1-9}$$

其中

$$|\varphi|^2_{H_{00}^{\frac{1}{2}}(E)} = \int_E\int_E \frac{(\varphi(x)-\varphi(y))^2}{|x-y|^2}\mathrm{d}s(x)\mathrm{d}s(y) + \int_E \frac{\varphi^2(x)}{\mathrm{dist}(x,\partial E)}\mathrm{d}s(x).$$

现给出几个重要的不等式.

(1) 庞加莱(Poincaré) 不等式：对任意 $u \in H^1(\Omega)$，有

$$d^{-1}|u|_{0,\Omega} \leqslant C\left(|u|^2_{1,\Omega} + d^{-2}\left|\int_{\partial\Omega} u\mathrm{d}s\right|^2\right)^{\frac{1}{2}}, \tag{1-10}$$

其中常数 C 与 u 及 d 无关(下同)，$|u|_{1,\Omega} = \left(\int_\Omega |\nabla u|^2\mathrm{d}x\right)^{\frac{1}{2}}$.

(2) 弗里德里希斯(Friedrichs) 不等式：对任何 $u \in H^1(\Omega)$，有

$$d^{-1}\|u-\gamma(u)\|_{0,\Omega} \leqslant C|u|_{1,\Omega}, \tag{1-11}$$

其中 $\gamma(u)$ 是 u 在 Ω 上的积分平均 $\gamma_\Omega(u)$，或是 u 在 $\Gamma_0 \subset \partial\Omega$ ($|\Gamma_0| = \mathrm{meas}(\Gamma_0) > 0$) 上的积分平均 $\gamma_{\Gamma_0}(u)$，即

$$\gamma_\Omega(u) = \frac{1}{|\Omega|}\int_\Omega u\mathrm{d}x,\quad \gamma_{\Gamma_0}(u) = \frac{1}{|\Gamma_0|}\int_{\Gamma_0} u\mathrm{d}s.$$

(3) 内插不等式：对任何 $\varphi \in H^1(\partial\Omega)$，有

$$\|\varphi\|_{\frac{1}{2},\partial\Omega} \leqslant Cd^{-\frac{1}{4}}\|\varphi\|^{\frac{1}{2}}_{0,\partial\Omega}\left(|\varphi|^2_{1,\partial\Omega} + d^{-1}\|\varphi\|^2_{0,\partial\Omega}\right)^{\frac{1}{4}}. \tag{1-12}$$

(4) ε- 不等式：对任何 $u \in H^1(\Omega)$，有(参见文献[20])

$$\|u\|_{0,\partial\Omega} \leqslant C(\varepsilon^{-1}\|u\|_{0,\Omega} + \varepsilon|u|_{1,\Omega}),\quad \varepsilon \in (0,1). \tag{1-13}$$

(5) 迹不等式：对任何 $u \in H^1(\Omega)$，有

$$\|u\|_{\frac{1}{2},\partial\Omega} \leqslant C\left(|u|^2_{1,\Omega} + d^{-2}\|u\|^2_{0,\Omega}\right)^{\frac{1}{2}},\quad |u|_{\frac{1}{2},\partial\Omega} \leqslant C|u|_{1,\Omega}. \tag{1-14}$$

1.3　有限元空间及性质

设 $T^h = \{e_i\}$ 是 Ω 的拟一致、正规三角形或四边形剖分，直径为 h. 定义

$$V_h = \{v \mid v \in C^1(\Omega),\ v|_{e_i} \in P_1(e_i),\ \forall e_i \in T^h\},$$

其中 $P_1(e_i)$ 表 e_i 上的线性或双线性函数全体.

以“$\approx$”表范数等价，N_h 表 Ω 上的结点集，则对任何 $v \in V_h$，有

$$\|v\|^2_{0,\Omega} \approx h^2\sum_{x_i \in N_h} v^2(x_i),$$

$$|v|^2_{1,\Omega} \approx \sum_{e\in T^h}\sum_{x_i,x_j\in e\cap N_h}(v(x_i)-v(x_j))^2,$$

$$\|v\|^2_{\frac{1}{2},E} \approx h^2\sum_{\substack{x_i,x_j\in E\cap N_h\\ x_i\neq x_j}}\frac{(v(x_i)-v(x_j))^2}{|x_i-x_j|^2}.$$

上述式子的右端项就是通常的离散范数.

下列结果的证明见文献[8].

命题 4 若 $\varphi \in V_h|_E$,则有

$$\|\varphi\|_{0,\infty,E} \leqslant C\left(1+\log\frac{d}{h}\right)^{\frac{1}{2}}\|\varphi\|_{\frac{1}{2},E}. \tag{1-15}$$

下列结果可在文献[2],[20]或[32]中找到.

命题 5 若 $\varphi \in V_h|_{\partial\Omega}$,且 $\varphi \in H_{00}^{\frac{1}{2}}(E)$,则有

$$\|\varphi\|_{\frac{1}{2},\partial\Omega} \approx \|\varphi\|_{H_{00}^{\frac{1}{2}}(E)} \leqslant C\left(1+\log\frac{d}{h}\right)\|\varphi\|_{\frac{1}{2},E}. \tag{1-16}$$

定义双线性型

$$a(u,v)=\int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}x+\eta\int_\Omega uv\,\mathrm{d}x,\quad u,v\in H^1(\Omega),$$

其中 $\eta \geqslant 0$ 是一实数.

定义 5 设 $\varphi \in H^{\frac{1}{2}}(\partial\Omega)$,若 $u|_{\partial\Omega}=\varphi$,且满足

$$a(u,v)=0,\quad \forall v\in H_0^1(\Omega),$$

则称 u 是 φ 在 Ω 上的 **a 扩张函数**.特别,当 $\eta=0$ 时,称 u 是 φ 在 Ω 上的**调和扩张函数**.

定义 6 设 $\varphi_h \in V_h|_{\partial\Omega}$,若 $u_h|_{\partial\Omega}=\varphi_h$,且满足

$$a(u_h,v_h)=0,\quad \forall v_h\in V_h\cap H_0^1(\Omega),$$

则称 u_h 是 φ_h 在 Ω 上的 **a 离散扩张函数**.特别,当 $\eta=0$ 时,称 u_h 是 φ_h 在 Ω 上的**离散调和扩张函数**.

命题 6 若 u_h 是 $\varphi_h(\in V_h|_{\partial\Omega})$ 在 Ω 上的离散调和扩张,则

$$|u_h|_{1,\Omega} \leqslant C|\varphi_h|_{\frac{1}{2},\partial\Omega}. \tag{1-17}$$

2 非重叠区域分解法

顾名思义,非重叠区域分解就是将求解区域分解成若干个互不重叠的子区域,其思想类似于工程中的"子结构法",最初称为"容度矩阵方法",早期文章见文献[6],[7]和[26].

2.1 两子域情形

为叙述方便,考虑简单模型:

$$\begin{cases}-\Delta u+\eta u=f, & \text{在 } \Omega \text{ 内},\\ u=0, & \text{在 } \partial\Omega \text{ 上},\end{cases} \tag{2-1}$$

其中 $\Omega\subset\mathbf{R}^2$ 是由分段光滑的闭曲线所围成的有界开域,常数 $\eta\geqslant 0$,$f\in L^2(\Omega)$.

将 Ω 分解成 $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2, \Omega_1 \cap \Omega_2 = \varnothing$，记 $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. 以 $u \in H_0^1(\Omega)$ 表模型(2-1)的弱解，令 $u_k = u|_{\Omega_k}(k = 1,2)$，则 u_k 满足方程

$$\begin{cases} -\Delta u_k + \eta u_k = f, & \text{在 } \Omega_k \text{ 内}, \\ u_k = 0, & \text{在 } \partial\Omega \cap \partial\Omega_k \text{ 上}, \end{cases} \tag{2-2}$$

及界面条件

$$u_1|_\Gamma = u_2|_\Gamma, \tag{2-3}$$

$$\left.\frac{\partial u_1}{\partial n_1}\right|_\Gamma = -\left.\frac{\partial u_2}{\partial n_2}\right|_\Gamma, \tag{2-4}$$

其中 n_k 表 $\partial\Omega_k$ 的外法向向量.

条件(2-3)表明，可由 $\varphi = u_1|_\Gamma = u_2|_\Gamma$ 定义唯一的 $\varphi \in H^{\frac{1}{2}}(\Gamma)$. 假设 φ 已知，则 u_k 可通过并行求解狄利克雷(Dirichlet)问题

$$\begin{cases} -\Delta u_k + \eta u_k = f, & \text{在 } \Omega_k \text{ 内}, \\ u_k = 0, & \text{在 } \partial\Omega \cap \partial\Omega_k \text{ 上}, \\ u_k = \varphi, & \text{在 } \Gamma \text{ 上} \end{cases} \tag{2-5}$$

得到($k = 1,2$). 但实际上并不预先知道 φ.

为确定 φ，记 $u_k = u_k(\varphi, f)$，并定义斯德罗夫-庞加莱(Steklov-Poincaré)算子 $S_k: H^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma)$ 为

$$S_k\varphi = \left.\frac{\partial u_k(\varphi, 0)}{\partial n_k}\right|_\Gamma, \quad k = 1,2,$$

其中 $u_k(\varphi, 0)$ 即为 φ 在 Ω_k 内的标准扩张函数(方程(2-5)中取 $f = 0$ 时的解).

因 $u_k = u_k(\varphi, 0) + u_k(0, f)$，故由条件(2-4)得关于 φ 的界面方程

$$(S_1 + S_2)\varphi = g, \tag{2-6}$$

其中 $g = -\left[\frac{\partial u_1(0,f)}{\partial n_1} + \frac{\partial u_2(0,f)}{\partial n_2}\right]$，可预先(并行)求得.

定理1 界面算子 $\tilde{S} = S_1 + S_2$ 是对称、正定的，从而方程(2-6)在 $H^{\frac{1}{2}}(\Gamma)$ 上唯一可解.

以上定理的证明是直接的(参见文献[20]).

显然，方程(2-6)的解析解无法求得，下面给出其离散形式.

将 Ω_1 和 Ω_2 分别作标准的三角形单元剖分(直径为 h)，使其在 Γ 上的网格结点重合. 显然，它们的全体构成 Ω 的有限元剖分. 以 $\mathring{V}_h \subset C(\Omega)$ 表相应的分片线性元空间，方程(2-1)的有限元逼近 $u_h \in \mathring{V}_h$ 满足

$$\sum_{i=1}^{2} a_i(u_h, v_h) = (f, v_h), \quad \forall v_h \in \mathring{V}_h, \tag{2-7}$$

其中$(\cdot,\cdot)$表 Ω 上的 L^2 内积，

$$a_i(v, w) = \int_{\Omega_i} \nabla v \cdot \nabla w \mathrm{d}x + \eta \int_{\Omega_i} vw \mathrm{d}x, \quad v, w \in H_0^1(\Omega).$$

对 $\mathring{V}_h$ 的结点基函数，以 $x_i(i=1,2,3)$ 分别表示 u_h 对应于 Ω_1、Ω_2 和 Γ 的坐标向量. 以 K_{11}、K_{22} 和 K_{33} 分别表示由 Ω_1 内部的、Ω_2 内部的和 Γ 上的结点基函数得到的刚度矩阵. 则(2-7) 可写成

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^{\mathrm{T}} & K_{23}^{\mathrm{T}} & K_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}. \tag{2-8}$$

将内部变量 x_1、x_2 消去，方程(2-8) 约化为关于 x_3 的线性方程

$$S_h x_3 = b, \tag{2-9}$$

其中

$$S_h = K_{33} - K_{13}^{\mathrm{T}} K_{11}^{-1} K_{13} - K_{23}^{\mathrm{T}} K_{22}^{-1} K_{23},$$
$$b = f_3 - K_{13}^{\mathrm{T}} K_{11}^{-1} f_1 - K_{23}^{\mathrm{T}} K_{22}^{-1} f_2.$$

方程(2-9) 就是界面方程(2-6) 的代数形式，S_h 恰为 $\widetilde{S}$ 的刚度矩阵. 通常将 S_h 称为**舒尔余矩阵**(Schur complement matrix).

直接解方程(2-9) 存贮量较大，为此定义

$$S_h^{(2)} = K_{33}^{(2)} - K_{13}^{\mathrm{T}} K_{11}^{-1} K_{13},$$

其中 $K_{33}^{(2)}$ 表 K_{33} 中对应于 Ω_2 的部分(即只在 Ω_2 上积分). $S_h^{(2)}$ 恰为 S_2 的刚度矩阵.

以 $S_h^{(2)}$ 作为 S_h 的预条件子，用里查德森迭代求解方程(2-9).

算法 1 取定初始向量 $x_3^{(0)}$，若已求得 $x_3^{(i)}$，则按下列公式计算 $x_3^{(i+1)}$：

$$x_3^{(i+1)} = x_3^{(i)} + \theta_i (S_h^{(2)})^{-1} (b - S_h x_3^{(i)}), \tag{2-10}$$

其中 $0 < \theta_i < 1$ 为松弛因子.

下列结果是基本的(参见文献[20]).

定理 2 迭代矩阵 $(S_h^{(2)})^{-1} S_h$ 的条件数与网格参数 h 无关，即其上界是一常数.

该定理表明，对适当选取的松弛因子 θ_i，算法 2-1 收敛得较快.

为避免精确求解 K_{11}^{-1} 和 K_{22}^{-1}，通常采用另一种与(2-10) 式等价的迭代方法. 记

$$\Phi_h = \{v|_\Gamma \mid v \in \mathring{V}_h\} \subset H_{00}^{\frac{1}{2}}(\Gamma);$$
$$V_h^i = \{v|_{\overline{\Omega}_i} \mid v \in \mathring{V}_h\}, \quad i = 1,2;$$
$$\mathring{V}_h^i = V_h^i \cap H_0^1(\Omega_i), \quad i = 1,2.$$

算法 2 离散 D-N 交替法

选初值 $\varphi_h^0 \in \Phi_h$. 若已求得 φ_h^n，则按下列方式计算 φ_h^{n+1}：

步 1 在 Ω_1 上解离散狄利克雷问题：求 $u_{h_1}^n \in V_h^1$，使

$$\begin{cases} a_1(u_{h_1}^n, v_h) = \int_{\Omega_1} f v_h \mathrm{d}x, & \forall v_h \in \mathring{V}_h^1, \\ u_{h_1}^n = \varphi_h^n, & \text{在 } \Gamma \text{ 上}. \end{cases}$$

步2 在 Ω_2 上解离散诺伊曼(Neumann)问题:求 $u_{h_2}^n \in V_h^2$,使

$$a_2(u_{h_2}^n, v_h) = \int_{\Omega_2} f v_h \mathrm{d}x + \int_\Gamma \frac{\partial u_{h_1}{}^n}{\partial n_2} v_h \mathrm{d}s, \quad \forall v_h \in V_h^2.$$

步3 令 $\varphi_h^{n+1} = \theta_n u_{h_2}^n |_\Gamma + (1-\theta_n)\varphi_h^n, 0 < \theta_n < 1$.

显然,在求得 φ_h^{n+1} 的同时,也已求得了 u_k 的近似 $u_{h_k}{}^{n+1}(k = 1,2)$.

下述结果的证明可在文献[20]中找到.

定理3 算法2与算法1相互等价.

注2.1:由上章命题1知,松弛因子 θ_i 的选取对算法1(或算法2)的收敛速度影响甚大.除少数特殊问题外,$(S_h^{(2)})^{-1}S_h$ 的最小、最大特征值难以精确估计,故难以直接选到好的 θ_i.一种变动方式是,将其与最小余量法结合起来选取 θ_i(详见文献[20]).

注2.2:从"将大化小"角度看,分成两个子区域显然是不够的.但对某些实际问题,如有限元与边界元相耦合的问题,它可起到"化复杂为简单"的作用(参见文献[33]).

2.2 多子域情形

在工程应用中,通常是将求解区域分解成多个子区域.本节仍考虑模型方程(2-1).

将 Ω 分解成 N 个子区域:$\overline{\Omega} = \bigcup_{i=1}^N \overline{\Omega}_i$.假定:

1° 当 $i \neq j$ 时,$\Omega_i \cap \Omega_j = \varnothing$;

2° 当 Ω_i 与 Ω_j 相邻时,$\partial\Omega_i \cap \partial\Omega_j$ 是 Ω_i 和 Ω_j 的一条公共边 Γ_{ij}.记 $\Gamma = \bigcup \Gamma_{ij}$,$\Gamma$ 称为界面;

3° 每个 Ω_i 的"尺寸"为 d,即存在常数 c_0 和 C_0,使 Ω_i 包含(包含在)一个直径为 $c_0 d(C_0 d)$ 的圆(圆内).

为突出其基本思想,仍先讨论连续问题.

方程(2-1)的弱解 $u \in H_0^1(\Omega)$ 满足

$$\sum_{i=1}^N a_i(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega), \tag{2-11}$$

其中 $a_i(\cdot,\cdot)$ 按上节的方式定义.

令 $u_k = u|_{\Omega_k}$,则方程(2-11)等价于联立方程组

$$\begin{cases} a_k(u_k, v) = \int_{\Omega_k} f v \mathrm{d}x, & \forall v \in H_0^1(\Omega_k), k = 1,\cdots,N, \\ u_k = 0, & \text{在 } \partial\Omega \cap \partial\Omega_k \text{ 上}, \\ u_i = u_j, & \text{在 } \Gamma_{ij} \text{ 上}, i < j, \\ \dfrac{\partial u_i}{\partial n_i} = -\dfrac{\partial u_j}{\partial n_j}, & \text{在 } \Gamma_{ij} \text{ 上}, \end{cases} \tag{2-12}$$

其中 n_i 仍表 $\partial\Omega_i$ 的外法向向量.

定义 $\varphi\in H^{\frac{1}{2}}(\Gamma)$,使 $\varphi|_{\Gamma_{ij}}=u_i|_{\Gamma_{ij}}=u_j|_{\Gamma_{ij}}$.与上节类似地可得到关于 φ 的界面方程

$$\left(\sum_{k=1}^{N}S_k\right)\varphi=g,\tag{2-13}$$

其中 Ω_k 上的斯德罗夫-庞加莱算子 $S_k:H^{\frac{1}{2}}(\Gamma)\to H^{-\frac{1}{2}}(\partial\Omega_k)$ 定义为

$$S_k\varphi=\left.\frac{\partial u_k(\varphi,0)}{\partial n_k}\right|_{\partial\Omega_k},$$

函数

$$g=-\sum_{i=1}^{N}\left.\frac{\partial u_k(0,f)}{\partial n_k}\right|_{\partial\Omega_k},$$

而 $u_k(\varphi,0)$ 是 φ 在 Ω_k 内的 a_k 扩张函数,$u_k(0,f)$ 是方程(2-12)取零边值时的解.

同两子域情形,可证明(参见文献[31])

定理4　界面算子 $\widetilde{S}=\sum_{k=1}^{N}S_k$ 在 $H^{\frac{1}{2}}(\Gamma)$ 上对称、正定,从而方程(2-13)唯一可解.

通常将 φ 称作**界面变量**.假定 Ω 是多角形区域,将 $\widetilde{\Omega}_i$ 作直径为 h 的拟一致、正规三角形或四边形剖分,使 Ω_i 与 Ω_j 在其公共边 Γ_{ij} 上的结点重合,单元集记作 T_h^i.显然,它们的全体(连同 Ω_i 的顶点)构成 Ω 的剖分.

定义1　初始剖分的结点,即所有 Ω_i 的顶点(在 Ω 内),称为**内交点**.

以 $P_1(e)$ 表单元 e 上的线性或双线性函数的全体.令

$$V_h^i=\{v\mid v\in C(\Omega_i),v|_{\partial\Omega}=0,v|_e\in P_1(e),\forall e\in T_h^i\},$$

$$\mathring{V}_h^i=\{v\mid v\in V_h^i,v|_{\partial\Omega_i}=0\},$$

$$V_h=\{v\mid v\in C(\Omega),v|_{\Omega_i}\in V_h^i,\text{对每个 } i\}.$$

方程(2-1)的有限元逼近 $u_h\in V_h$ 满足

$$\sum_{i=1}^{N}a_i(u_h,v_h)=(f,v_h),\quad\forall v_h\in V_h.\tag{2-14}$$

由于内交点联系着多个子区域,故不便按上节的方式导出方程(2-13)的离散形式.

记 $u_{hi}=u_h|_{\Omega_i}$,方程(2-14)等价于联立方程组

$$\begin{cases}a_k(u_{hk},v_h)=\int_{\Omega_k}fv_h\mathrm{d}x,\quad\forall v_h\in\mathring{V}_h^k,k=1,\cdots,N,\\ u_{hk}=0,\qquad\text{在 }\partial\Omega\cap\partial\Omega_k,\\ u_{hi}=u_{hj},\qquad\text{在 }\Gamma_{ij}\text{ 上},i<j,\\ \int_{\Gamma_{ij}}\frac{\partial u_{hi}}{\partial n_i}v_h\mathrm{d}s+\int_{\Gamma_{ij}}\frac{\partial u_{hj}}{\partial n_j}v_h\mathrm{d}s=0,\quad\forall v_h\in V_h,\end{cases}\tag{2-15}$$

以 $T_{ri}: H^1(\Omega_i) \to H^{\frac{1}{2}}(\partial\Omega_i)$ 表通常的迹算子,令

$$V(\partial\Omega_i) = \{\psi \mid 存在 v \in V_h^i, 使 \psi = T_{ri}v\}.$$

定义

$$V(\Gamma_{ij}) = \{\psi \mid 存在 v \in V(\partial\Omega_i), 使 \psi = v|_{\Gamma_{ij}}\}$$

及界面空间

$$V(\Gamma) \subset \{\psi \mid \psi \in C(\Gamma), \psi|_{\Gamma_{ij}} \in V(\Gamma_{ij})\}.$$

定义离散 a_i 扩张算子 $T_{ri}^{-1}: V(\partial\Omega_i) \to V_h^i$ 如下:

$$\begin{cases} a_i(T_{ri}^{-1}\varphi_i, v) = 0, & \forall v \in \mathring{V}_h^i, \\ T_{ri}^{-1}\varphi_i = \varphi_i, & 在 \partial\Omega_i 上, \quad \varphi_i \in V(\partial\Omega_i). \end{cases}$$

定义 Ω_i 上的离散算子 $S_h^{(i)}: V(\partial\Omega_i) \to (V(\partial\Omega_i))^*$ 为

$$\langle S_h^{(i)}\varphi_i, \psi_i\rangle = a_i(T_{ri}^{-1}\varphi_i, T_{ri}^{-1}\psi_i), \varphi_i, \psi_i \in V(\partial\Omega_i),$$

其中$\langle\cdot,\cdot\rangle$ 表 $\partial\Omega_i$ 上(也表 Γ 上)的 L^2 内积.

定义 $\varphi_h \in V(\Gamma)$,使 $\varphi_h|_{\Gamma_{ij}} = u_{hi}|_{\Gamma_{ij}} = u_{hj}|_{\Gamma_{ij}}$. 以 $I_i: V(\Gamma) \to V(\partial\Omega_i)$ 表自然的限制算子. 由方程(2-14) 和界面条件(离散形式)(2-15) 可得到 φ_h 的界面方程(参见文献[14] 和[31])

$$\left(\sum_{i=1}^{N} I_i^T S_h^{(i)} I_i\right)\varphi_h = -\sum_{i=1}^{N} I_i^T T_{ri}^{-T} f, \quad \varphi_h \in V(\Gamma), \tag{2-16}$$

其中 I_i^T 表 I_i 的共轭算子(即零扩张算子),T_{ri}^{-T} 表 T_{ri}^{-1} 在内积$\langle\cdot,\cdot\rangle$ 下的共轭算子.

按标准的方式可证明下面的定理.

定理5 **离散界面算子** $S_h = \sum_{i=1}^{N} I_i^T S_h^{(i)} I_i$ 在 $V(\Gamma)$ 上对称、正定,从而方程(2-16) 唯一可解.

现给出方程(2-16) 的代数形式.

以$\{\varphi_k^i\}_{k=1}^{N_i}$,$\{\psi_k^i\}_{k=1}^{M_i}$ 分别表 $\mathring{V}_h^i$ 和 $V(\partial\Omega_i)$ 中的结点基函数集,记

$$a_{jk}^i = a_i(\varphi_j^i, \varphi_k^i), \quad b_{jk}^i = a_i(\varphi_j^i, \psi_k^i), \quad c_{jk}^i = a_i(\psi_j^i, \psi_k^i),$$
$$f_j^i = (f, \varphi_j^i)_{\Omega_i}, \quad g_j^i = (f, \psi_j^i)_{\Omega_i},$$

其中$(\cdot,\cdot)_{\Omega_i}$ 表 Ω_i 上的 L^2 内积.

定义矩阵

$$A_i = [a_{jk}^i]_{jk}, \quad B_i = [b_{jk}^i]_{jk}, \quad C_i = [c_{jk}^i]_{jk}.$$

和向量

$$f_i = (f_1^i, f_2^i, \cdots, f_{N_i}^i)^T, \quad g_i = (g_1^i, g_2^i, \cdots, g_{M_i}^i)^T.$$

以 $\bar{I}_i$ 表对应于算子的0-1符号矩阵,X 表 φ_h 在 $V(\Gamma)$ 基下的坐标向量. 易推得方程(2-16) 的代数形式为(参见文献[31])

$$\left(\sum_{i=1}^{N} \bar{I}_i^T \bar{S}_i \bar{I}_i\right)X = b, \tag{2-17}$$

其中

$$\tilde{S}_i = (C_i - B_i^T A_i^{-1} B_i), \quad b = \sum_{i=1}^{N} \bar{I}_i^T (g_i - B_i^T A_i^{-1} f_i).$$

记 $\bar{S} = \sum_{i=1}^{N} \bar{I}^T \tilde{S}_i \bar{I}_i$. 由于界面矩阵 $\bar{S}$ 的结构复杂，而且其阶数一般较大，故(2-17)式不宜直接求解. 在构造其预条件子之前，先介绍一种迭代方法.

从上一节可知，离散 D-N 交替法并不需建立界面方程. 自然，希望能将其思想推广到多子域情形，但遗憾的是，在程序实现时难以确定何处取狄利克雷边界条件，何处取诺伊曼边界条件. 为此，文献[19]提出并分析了另一类迭代方法.

算法 3 多子域迭代法

步 1 对每个 i，取定 $u_{hi}^0 \in V_h^i$，并计算 $\left.\frac{\partial u_{hi}^0}{\partial n_i}\right|_{\partial\Omega_i}$.

步 2 在每个子区域上(并行)求解第三类边值问题

$$\begin{cases} a_i(u_{hi}^n, v_i) + \rho(u_{hi}^n, v_i) = (f, v_i)_{\Omega_i} + \\ \qquad \sum_j \int_{\Gamma_{ij}} \left(\frac{\partial u_{hj}^{n-1}}{\partial n_j} + \rho u_{hj}^{n-1}\right) v_i \mathrm{d}s, \quad \forall v_i \in V_h^i, \\ u_{hi}^n = 0, \qquad \text{在 } \partial\Omega_i \cap \partial\Omega, \end{cases}$$

其中 $\rho > 0$ 为松弛因子.

步 3 $n := n + 1$，转步 2.

关于其收敛性，有下列结果(见文献[19]).

定理 6 对任意的 $\rho > 0$，算法 3 收敛，且 u_{hi}^n 的极限恰为 $u_{hi}(i = 1, \cdots, N)$.

注 2.3: 算法 3 的收敛速度与 ρ 的值密切相关，而且一般收敛得很慢. 因此，常用的还是"子结构"型方法(即须求解界面方程).

2.3 界面预条件子的构造

从上两节可看出，非重叠区域分解法(除迭代法)的基本思想是：引进适当的界面变量，通过并行求解若干子问题，将内部变量消去，得到界面变量的方程. 求解界面方程后，将界面变量的值回代即求得内部变量的值，进而得到原问题的逼近解. 能否快速地求解(离散)界面方程(2-16)或(2-17)，是衡量非重叠区域分解法好坏的标志. 因为一旦界面变量 φ_h 已求出，内部变量 u_{hi} 就可并行、独立地求得. 求解界面方程(2-16)最好的一种方法是预条件共轭梯度法. 因此，界面预条件子的构造是非重叠区域分解法研究和应用中的核心问题. 当然，这也是非常困难的问题.

命题 1 离散界面算子 S_h 的条件数为

$$\mathrm{cond}(S_h) = O(h^{-1} d^{-1}).$$

该命题表明，$\mathrm{cond}(S_h)$ 一般很大. 因此，构造 S_h 的预条件子是必要的.

下面从一个一般框架出发，介绍 S_h 的两类重要预条件子. 为简化符号，将下标

"h"省略,即把方程(2-16)写成

$$S\varphi = g, \quad \varphi \in V(\Gamma). \tag{2-18}$$

将界面空间 $V(\Gamma)$ 分解为 $V(\Gamma) = V_0 + \overline{V}$,其中 V_0、$\overline{V} \in V(\Gamma)$. 又将 Γ 分解成 $\Gamma = \bigcup_{k=1}^{p} \Gamma_k$,其中 Γ_k 是Γ的子集,设 $V_k \subset V(\Gamma_k) = V(\Gamma)\mid_{\Gamma_k}$. 定义内射 $D_k: V(\Gamma_k) \to \overline{V}$,满足单位分解条件:

$$\sum_{k=1}^{p} D_k I_k = I, \quad 在\ \overline{V}\ 上, \tag{2-19}$$

其中 $I_k: \overline{V} \to V_k$ 是自然的限制算子,I 为 $\overline{V}$ 上的单位算子. 设存在算子 $r_k: V(\Gamma_k) \to V_k$,使对任何 $\psi_k \in V(\Gamma_k)$,有 $D_k r_k \psi_k = D_k \psi_k$. 以 $D_k^T: \overline{V} \to V(\Gamma_k)$ 表 D_k 在 L^2 内积 $\langle\cdot,\cdot\rangle$ 下的共轭算子,它满足条件:对 $\psi \in \overline{V}$,有 $D_k^T\psi \in V_k$. 设 $S_k: V_k \to V_k(k = 0, 1, \cdots, p)$ 为对称正定算子,以 Q_0、$\overline{Q}$ 表从 $V(\Gamma)$ 到 V_0 和 $\overline{V}$ 上的$\langle\cdot,\cdot\rangle$正交投影算子. 定义 S 的预条件子为

$$M^{-1} = S_0^{-1}Q_0 + \left(\sum_{k=1}^{p} D_k S_k^{-1} D_k^T\right)\overline{Q}. \tag{2-20}$$

下列结果提供了估计条件数 $\mathrm{cond}(M^{-1}S)$ 的基本思路(证明见文献[14]).

定理 7 若下列条件满足:

1° 对任意 $\varphi \in V(\Gamma)$,存在分解 $\varphi = \varphi_0 + \overline{\varphi}$,其中 $\varphi_0 \in V_0$,$\overline{\varphi} \in \overline{V}$,使

$$\langle S_0\varphi_0, \varphi_0\rangle + \sum_{k=1}^{p}\langle S_k r_k I_k\overline{\varphi}, r_k I_k\overline{\varphi}\rangle_{\Gamma_k} \leqslant \alpha_1\langle S\varphi, \varphi\rangle,$$

2° 对任意 $\varphi_k \in V_k(k = 0, 1, \cdots, p)$,有

$$\left\langle S\left(\sum_{k=1}^{p} D_k\varphi_k + \varphi_0\right), \sum_{k=1}^{p} D_k\varphi_k + \varphi_0\right\rangle \leqslant \alpha_2\sum_{k=0}^{p}\langle S_k\varphi_k, \varphi_k\rangle_{\Gamma_k},$$

则
$$\alpha_1^{-1}\langle\varphi, S\varphi\rangle \leqslant \langle M^{-1}S\varphi, S\varphi\rangle \leqslant \alpha_2\langle\varphi, S\varphi\rangle, \quad \forall\,\varphi \in V(\Gamma),$$

即
$$\mathrm{cond}(M^{-1}S) \leqslant \alpha_1\alpha_2.$$

注 2.4: 算子 r_k 的作用仅是为了使 $r_k I_k\overline{\varphi} \in V_k$,因为 S_k 仅在 V_k 上可逆. 若 $V_k = \overline{V}\mid_{\Gamma_k}$,则可取 $r_k = I$.

先介绍第一类预条件子,它是文献[2]提出的(称为 BPS 预条件子,对应于 $\eta = 0$).

将 Γ 分成 $\Gamma = \bigcup_{i<j}\Gamma_{ij}$,相应的局部界面空间记作 $V_{ij} = \overline{V}\mid_{\Gamma_{ij}}$(因此取 $r_{ij} = I$). 限制算子记作 I_{ij},局部可解子记作 S_{ij}. 这种分解的一个重要特点是 Γ_{ij} 互不重叠,因此将内射 D_{ij} 取作自然的零扩张算子 I_{ij}^T,就使单位分解条件(2-19)满足. 此时,S 的预条件子(2-20)可写成

$$M_1^{-1} = S_0^{-1}Q_0 + \left(\sum_{i<j} I_{ij}^T S_{ij}^{-1} I_{ij}\right)\overline{Q}. \tag{2-21}$$

需给出空间 V_0、V_{ij} 和算子 S_0、S_{ij} 的定义. 记

$$V^0(\Gamma_{ij}) = \{\psi \mid \psi \in V(\Gamma_{ij}), \psi\ 在\ \Gamma_{ij}\ 的端点上为零\}.$$

定义

$$V_0 = \{\psi \mid \psi \in V(\Gamma), \psi|_{\Gamma_{ij}} \text{为线性函数}\},$$

$$V_{ij} = \{\psi \in V^0(\Gamma_{ij}) \mid \text{存在 } \psi_\Gamma \in V(\Gamma_{ij}) \text{ 和 } \psi_0 \in V_0, \text{使 } \psi = \psi_\Gamma - \psi_0\},$$

$$\langle S_0\varphi_0, \psi_0\rangle = \sum_{\Gamma_{ij}} (\varphi_0(v_i) - \varphi_0(v_j))(\psi_0(v_i) - \psi_0(v_j)), \quad \varphi_0, \psi_0 \in V_0,$$

其中 v_i、v_j 是 Γ_{ij} 的两个端点.

以 $l_0: V^0(\Gamma_{ij}) \to V^0(\Gamma_{ij})$ 表如下离散算子

$$\langle l_0\varphi, \psi\rangle_{\Gamma_{ij}} = \int_{\Gamma_{ij}} \nabla\varphi \cdot \nabla\psi \mathrm{d}s, \quad \forall \varphi, \psi \in V^0(\Gamma_{ij}),$$

定义 $S_{ij} = l_0^{\frac{1}{2}}$.

定理 8 对预条件子 M_1,有

$$\mathrm{cond}(M_1^{-1}S) \leqslant C(1 + \log^2\frac{d}{h}).$$

注 2.5: 习惯上将 V_0 称为**粗子空间**,S_0 称为**粗可解子**.它在界面预条件子的构造中起关键作用.

设 $\hat{V} \subset V(\Gamma)$,称 $\hat{S}: \hat{V} \to \hat{V}$ 是 S 在 $\hat{V}$ 上的限制算子,如果

$$\langle \hat{S}\hat{\varphi}, \hat{\psi}\rangle = \langle S\hat{\varphi}, \hat{\psi}\rangle, \quad \forall \hat{\varphi}, \hat{\psi} \in \hat{V}.$$

注 2.6: 在 M_1^{-1} 的定义中,若将 S_0 和 S_{ij} 分别取作 S 在 V_0 和 I_{ij}^T 在 V_{ij} 上的限制算子,定理 2-8 仍成立.以上定义的 S_0 和 S_{ij} 分别与其谱等价,这样定义可使计算更为简便.此外,若直接定义 $V_{ij} = V^0(\Gamma_{ij})$,定理 2-8 同样成立.

关于 M_1^{-1} 的算法实施可参见文献[32]或下一节.

再介绍第二类预条件子,它是文献[21]提出的(习惯上称其为**平衡预条件子**,对应于 $\eta = 0$).将 Γ 分成 $\Gamma = \bigcup_i \partial\Omega_i$,相应的局部界面空间记作 V_i,局部可解子记作 $\tilde{S}_i$.预条件子可写成

$$M_2^{-1} = S_0^{-1}Q_0 + \left(\sum_{i=1}^{N} D_i \tilde{S}_i^{-1} D_i^T\right)\bar{Q} \tag{2-22}$$

首先指出,$\tilde{S}_i$ 不能取上节定义的 $S_h^{(i)}$,因为它不可逆($\eta = 0$);D_i 也不能取自然的零扩张算子 I_i^T,因为分解 $\Gamma = \bigcup_i \partial\Omega_i$ 是重叠的,单位分解条件(2-19)不满足.

定义 V_i 如下:若 $\mathrm{null}(S_h^{(i)}) \neq \{0\}$,则

$$V_i = \{\psi \mid \psi \in V(\partial\Omega_i), \gamma_{\partial\Omega_i}(\psi) = 0\},$$

其中 $\gamma_{\partial\Omega_i}(\psi)$ 表 ψ 在 $\partial\Omega_i$ 上的积分平均;否则由 $V_i = V(\partial\Omega_i)$ 易知,$S_h^{(i)}: V_i \to V_i$ 是可逆的,其逆记作 $S_i^\dagger$(称为广义逆).定义 $\tilde{S}_i^{-1} = S_i^\dagger$.

定义记数函数

$$\nu = \sum_{i=1}^{N} \chi_{\partial\Omega_i}(x), \quad x \in \Gamma,$$

及内射 $\theta_i: V(\partial\Omega_i) \to V(\Gamma)$

$$\theta_i\psi = \begin{cases} \nu^{-1}\psi, & 在\ \partial\Omega_i\ 的结点上, \\ 0, & 在\ \Gamma \setminus \partial\Omega_i\ 的结点上 \end{cases} \qquad (\psi \in V(\partial\Omega_i)).$$

粗子空间 V_0 定义为

$$V_0 = \text{span}\{\theta_i 1 \mid 对满足\ \partial\Omega_i \cap \partial\Omega = \varnothing\ 的所有\ i\}.$$

以 $\overline{V}$ 表 V_0 在内积 $\langle\cdot,\cdot\rangle$ 下的正交补. 以 $\mathrm{P}_0: V(\Gamma) \to V_0$ 表在内积 $\langle S\cdot,\cdot\rangle$ 下的正交投影算子, 记 $\mathrm{P}_0^{\perp} = \mathrm{I} - \mathrm{P}_0$. 原来的平衡预条件子定义为(见文献[21],[32])

$$\mathrm{M}_b^{-1} = \mathrm{P}_0\mathrm{S}^{-1} + \mathrm{P}_0^{\perp}\left(\sum_{i=1}^{N} \mathrm{Q}_i\mathrm{S}_i^{\dagger}\mathrm{Q}_i^{\mathrm{T}}\right)\mathrm{S}\mathrm{P}_0^{\perp}\mathrm{S}^{-1}.$$

以下说明 M_b^{-1} 可归结为一般框架(2-20), 对 $\psi_i \in V(\partial\Omega_i)$, 定义 $\mathrm{r}_i\psi_i = \psi_i - \gamma_{\partial\Omega_i}(\psi_i)$. 记 $\mathrm{D}_i = (\mathrm{I} - \mathrm{P}_0)\mathrm{Q}_i$, 则易知 $\mathrm{D}_i: V_i \to \overline{V}$ 满足条件(2-19), 且 $\mathrm{D}_i\mathrm{r}_i\psi_i = \mathrm{D}_i\psi_i$, $\mathrm{D}_i^{\mathrm{T}}v \in V_i(v \in \overline{V})$. S_0 取作 S 在 V_0 上的限制算式.

以下定理的证明见文献[14]或[15].

定理 9　平衡预条件子 M_b^{-1} 可写成

$$\mathrm{M}_b^{-1} = \mathrm{S}_0^{-1}\mathrm{Q}_0 + \left(\sum_{i=1}^{N}\mathrm{D}_i\ \widetilde{\mathrm{S}}_i^{-1}\mathrm{D}_i^{\mathrm{T}}\right)\overline{\mathrm{Q}},$$

且有

$$\mathrm{cond}(\mathrm{M}_b^{-1}\mathrm{S}) \leqslant C\left(1 + \log^2\frac{d}{h}\right).$$

算子 M_b^{-1} 的行为可由下列算法刻画(见文献[21],[32]).

算法 4

对任意 $g \in V(\Gamma)$, $\varphi = \mathrm{M}_b^{-1}g$ 可按下列方式求得.

步 1　解粗问题, 求 $\widetilde{\varphi}_0 \in V_0$,

$$\langle \mathrm{S}\widetilde{\varphi}_0, \psi_0\rangle = \langle g, \psi_0\rangle, \forall\, \psi_0 \in V_0,$$

计算余量 $r_0 = g - \mathrm{S}\widetilde{\varphi}_0$.

步 2　并行求 $\varphi_i \in V_i$ 满足$(i = 1,\cdots,N)$

$$\langle \mathrm{S}\varphi_i, \psi_i\rangle = \langle r_0, \theta_i\psi_i\rangle, \quad \forall\, \psi_i \in V_i.$$

步 3　再解粗问题, 平衡余量 $r = g - S\sum_{i=1}^{N}\theta_i\varphi_i$,

$$\langle \mathrm{S}\varphi_0, \psi_0\rangle = \langle r, \psi_0\rangle, \quad \forall\, \psi_0 \in V_0.$$

步 4　计算 $\varphi = \varphi_0 + \sum_{i=1}^{N}\theta_i\varphi_i$.

注 2.7: 这两类预条件子各有优缺点, 第一类预条件子计算量小, 但程序难以实现(需处理内交点); 第二类预条件子程序较易实现, 但计算量增大了(在每个边上需求解两次).

2.4 非重叠区域分解预条件子

上节介绍的是构造非重叠区域分解界面问题的预条件子，本节则介绍利用非重叠区域分解法直接构造原问题的预条件子.

为体现非重叠区域分解法的特点，本节考虑稍不同的模型

$$\begin{cases} -\sum_{i,j=1}^{2}\frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right)=f, & \text{在 } \Omega \text{ 内}, \\ u=0, & \text{在 } \partial\Omega \text{ 上}, \end{cases} \tag{2-23}$$

其中 a_{ij} 是 Ω 上的函数，使 $[a_{ij}]$ 为对称一致正定矩阵；$\Omega\subset\mathbf{R}^2$ 是多边形区域.

按上一节的方式作区域剖分，并沿用有关记号. 定义双线性型

$$A(u,v)=\sum_{i,j=1}^{2}\int_{\Omega}a_{ij}\frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j}\mathrm{d}x,\quad u,v\in H^1(\Omega).$$

方程(2-23)的有限元问题是：求 $u_h\in V_h$，使

$$A(u_h,v)=(f,v),\quad \forall v\in V_h. \tag{2-24}$$

定义新的双线性型

$$\widetilde{A}(u,v)=\sum_{i=1}^{N}\widetilde{A}_i(u,v),\quad u,v\in H^1(\Omega),$$

其中

$$\widetilde{A}_i(u,v)=q_i\int_{\Omega_i}\nabla u\cdot\nabla v\mathrm{d}x,\quad u,v\in H^1(\Omega_i),$$

而 $q_i\in[q_0,q_1]$ 是常数，q_0、q_1 表二阶矩阵 $[a_{ij}(x)]$ $(x\in\Omega_i)$ 的最小、最大特征值.

如此定义的 $\widetilde{A}(u,v)$ 与 $A(u,v)$ 谱等价，但算法更易实施(详见[2]或[20]).

对任意 $w\in V_h$，作正交分解 $w=w_P+w_H$，其中 $w_P|_{\Omega_i}\in\mathring{V}_h^i$ 满足

$$\widetilde{A}(w_P,v)=\widetilde{A}(w,v),\ \forall v\in\mathring{V}_h^i(i=1,\cdots,N),$$

而 $w_H\in V_h$ 满足

$$\widetilde{A}(w_H,v)=0,\ \forall v\in\mathring{V}_h^i(i=1,\cdots,N),$$

即 w_H 由 $w|_\Gamma$ 在每个 Ω_i 上的离散调和扩张函数得到.

另一方面，定义粗子空间

$$V_d=\{v\mid v\in C(\Omega),v|_{\partial\Omega}=0,\text{对每个 }\Gamma_{ij}\text{ 有 }v|_{\Gamma_{ij}}\in P_1\},$$

以 $\widetilde{V}_h$ 表 V_h 的子空间，其中的函数在每个内交点处取零值. 显然，有直和分解

$$V_h=V_d+\widetilde{V}_h.$$

对应于此直和分解，有 $w_H=w_v+w_E$，其中 $w_v\in V_d$，$w_E\in\widetilde{V}_h$.

记 $\widetilde{V}(\Gamma_{ij})=\widetilde{V}_h|_{\Gamma_{ij}}$. 对每条边 Γ_{ij}，定义算子 l_0: $\widetilde{V}(\Gamma_{ij})\to\widetilde{V}(\Gamma_{ij})$ 为

$$\langle l_0\varphi,\psi\rangle_{\Gamma_{ij}} = \langle \varphi',\psi'\rangle_{\Gamma_{ij}},\quad \varphi,\psi\in \widetilde{V}(\Gamma_{ij}).$$

易知，l_0 是对称、正定的.

记 $\alpha_{ij} = q_i + q_j$. 定义 $A(\cdot,\cdot)$ 的预条件子为

$$B(w,\psi) = \widetilde{A}(w_P,\psi_P) + \frac{1}{2}\sum_{\Gamma_{ij}}\alpha_{ij}\langle l_0^{1/2}w_E,\psi_E\rangle_{\Gamma_{ij}} + \sum_{\Gamma_{ij}}\alpha_{ij}(w_v(v_i)-w_v(v_j))(\psi_v(v_i)-\psi_v(v_j)),\tag{2-25}$$

其中 v_i、v_j 表 Γ_{ij} 的两个顶点.

以下结果表明相应预条件迭代矩阵的条件数为 $O(1+\ln^2\frac{d}{h})$. 其证明参见文献[2]和[20].

定理 10　存在与网格参数 h 无关的常数 α_1、α_2，使

$$\alpha_1(1+\ln^2\frac{d}{h})B(w,w)\leqslant A(w,w)\leqslant \alpha_2 B(w,w),\quad \forall w\in V_h.$$

为了使预条件迭代简便，必须使方程

$$B(w,\psi) = (g,\psi),\quad \forall\psi\in V_h$$

能快速并行求解，文献[2]给出下列算法.

算法 5　BPS 预条件算法

步 1　并行求解子域上的狄利克雷问题：

$$\widetilde{A}_i(w_P,\psi) = (g,\psi),\ \forall\psi\in V_h,$$

得到 w_P.

步 2　并行求解 Γ_{ij} 上的一维问题.

$$\frac{1}{2}\alpha_{ij}\langle l_0^{1/2}w_E,\psi\rangle_{\Gamma_{ij}} = (g,\psi) - \widetilde{A}(w_P,\psi),\quad \forall\psi\in\widetilde{V}(\Gamma_{ij}).$$

得到 w_E（上式右端中的 ψ 已扩张到 $\widetilde{V}_h(\Omega)$ 上）.

步 3　解粗网格方程

$$\sum_{\Gamma_{ij}}\alpha_{ij}(w_v(v_i)-w_v(v_j))(\psi_v(v_i)-\psi_v(v_j)) = (g,\psi)-\widetilde{A}(w_P,\psi),\quad \forall\psi\in V_d,$$

得到 w_v 在内交点上的函数值.

步 4　以 $w_E + w_v$ 在 Γ_{ij} 上的值为边界条件，并行求各子域上的离散调和扩张函数，即解

$$\begin{cases}\widetilde{A}_i(w_H,\psi) = 0, & \forall\psi\in\mathring{V}_h,\\ w_H = w_E + w_v, & \text{在 }\Gamma_{ij}\text{ 上}\end{cases}\quad (i=1,\cdots,N),$$

得到 w_H.

步 5　令 $w = w_P + w_H$.

注 2.8：定义算子 l_0 的目的在于，当对 Γ_{ij} 实施等距剖分时，步 2 可快速求解（详

见文献[2] 和[20]).

注 2.9：把内交点处的值单独考虑,求解粗网格方程是本质的,否则定理 2-10 不成立.这正是多子域情形的复杂之处.

注 2.10：众所周知,原刚度矩阵(即 $A(\cdot,\cdot)$ 对应的刚度矩阵)的条件数为 $O(h^{-2})$.此外,算法 2-5 中的每个子问题都较小,且具高度的并行性.因此,预条件子 $B(\cdot,\cdot)$ 是非常有效的.

注 2.11：当方程(2-23)的系数 $a_{ij}(x)$ 在某些 Γ_{ij} 上间断(比如说是分片常数)时,本节结果(和上节结果)毫不受影响,这是非重叠区域分解法的优点.

注 2.12：预条件子(2-25)与上节介绍的 BPS 预条件子本质上是相同的,其差异在于,前者不需建立界面方程;而后者不需得到 $A(\cdot,\cdot)$ 对应的整个刚度矩阵.

2.5　非协调情形

非协调区域分解包括两种类型:非协调元的区域分解和非匹配网格区域分解.

先考虑第一种类型.尽管有各种不同的非协调元,但从区域分解角度看,其思想是一样的.这里仅以莫勒元(Morley element)为例介绍之(见文献[28]).

考虑重调和方程

$$\begin{cases}\Delta^2 u = f, & \text{在 } \Omega \text{ 内},\\ u = \partial_n u = 0, & \text{在 } \partial\Omega \text{ 上},\end{cases} \tag{2-26}$$

其中 Ω 是平面上的有界多角形区域;∂_n 表外法向导数(下同).

定义双线性型

$$a(u,v) = \sum_{|\alpha|=2}\int_\Omega D^\alpha u D^\alpha v \mathrm{d}x, \quad u,v \in H^2(\Omega),$$

则方程(2-26)的弱形式是:求 $u \in H_0^2(\Omega)$,使

$$a(u,v) = (f,v), \quad \forall v \in H_0^2(\Omega). \tag{2-27}$$

按 2.2 节的方式作非重叠区域分解,将每个子区域 Ω_i 作直径为 h 的三角形剖分(是拟一致、正规的),使任两个相邻子区域在公共边上的结点重合.它们的全体构成整个 Ω 上的三角剖分,单元集记作 T_h.

以 V_h 表**莫勒元空间**,即 V_h 中的函数在每个单元 $e \leqslant T_h$ 上为完全的二次多项式,其 6 个参数由 3 个顶点处的函数值和 3 边中点处的外法向导数值决定.V_h 中的函数跨过单元边时不保持连续性(在顶点处连续),故 $V_h \not\subset H^1(\Omega)$.以 $\mathring{V}_h$ 表 V_h 的子空间,其元素在 $\partial\Omega$ 上的参数值为零.

(2-27)式的有限元离散问题是:求 $u_h \in \mathring{V}_h$,使

$$a_h(u_h,v_h) = (f,v_h), \quad \forall v_h \in \mathring{V}_h. \tag{2-28}$$

其中

$$a_h(u,v) = \sum_{e\in T_h}\sum_{|\alpha|=2}\int_e D^\alpha u D^\alpha v \mathrm{d}x, \quad u,v \in V_h$$

按本章前几节的方式定义 Ω_i 上的局部莫勒元空间 V_h^i 和 $\mathring{V}_h^i(i = 1,\cdots,N)$.

定义局部双线性型

$$a_h^i(u,v) = \sum_{e\in T_h\cap\Omega_i}\sum_{|\alpha|=2}\int_e D^\alpha u D^\alpha v \mathrm{d}x, \quad u,v\in V_h^i.$$

对任意 $w\in \mathring{V}_h$,将其分解为 $w = w_P + w_H$,其中 $w_P\in \mathring{V}_h^1\oplus\mathring{V}_h^2\oplus\cdots\oplus\mathring{V}_h^N$,且满足

$$a_h^i(w_P,v) = a_h^i(w,v), \quad \forall v\in \mathring{V}_h^i(i = 1,\cdots,N),$$

$w_H\in \mathring{V}_h$ 满足

$$a_h^i(w_H,v) = 0, \quad \forall v\in \mathring{V}_h^i(i = 1,\cdots,N).$$

定义与 $\mathring{V}_h$ 相关的协调元空间

$$W_h = \{v\in C^1(\overline{\Omega}) \mid v|_e\in P_5(e),\forall e\in T_h, v = \partial_n v = 0 \text{ 在 } \partial\Omega \text{ 上}\}$$

和相应的协调插值算子 $\mathrm{E}_h:\mathring{V}_h\to W_h$(定义详见[28]).与上一节类似地,进一步将 w_H 分解成 $w_H = w_E + w_v$.记 $\overline{v} = \mathrm{E}_h v$(对 $v\in \mathring{V}_h$).

定义 $a_h(\cdot,\cdot)$ 的预条件子 $b_h(\cdot,\cdot)$ 如下:

$$\begin{aligned}b_h(w,\psi) = {} & a_h(w_P,\psi_P) + \sum_{\Gamma_{ij}}\{\langle\partial_s\overline{w}_E,\partial_s\overline{\psi}_E\rangle_{H_{00}^{\frac12}(\Gamma_{ij})} + \langle\partial_n\overline{w}_E,\partial_n\overline{\psi}_E\rangle_{H_{00}^{\frac12}(\Gamma_{ij})}\} + \\ & \sum_{\Gamma_{ij}}\{(w_v(v_i) - w_v(v_j) - D\overline{w}_v(v_i)(v_i - v_j))\cdot \\ & (\psi_v(v_i) - \psi_v(v_j) - D\overline{\psi}_v(v_i)(v_i - v_j))d^{-2} + \\ & (D\overline{w}_v(v_i) - D\overline{w}_v(v_j))(D\overline{\psi}_v(v_i) - D\overline{\psi}_v(v_j))\},\end{aligned}$$

其中 v_i、v_j 表 Γ_{ij} 的两个端点;内积 $\langle\cdot,\cdot\rangle_{H_{00}^{\frac12}(\Gamma_{ij})}$ 定义为

$$\begin{aligned}\langle\varphi,\psi\rangle_{H_{00}^{\frac12}(\Gamma_{ij})} = {} & \int_{\Gamma_{ij}}\int_{\Gamma_{ij}}\frac{(\varphi(x)-\varphi(y))(\psi(x)-\psi(y))}{|x-y|}\mathrm{d}s(x)\mathrm{d}s(y) + \\ & \int_{\Gamma_{ij}}\varphi(x)\psi(x)\left(\frac{1}{|x-v_i|}+\frac{1}{|x-v_j|}\right)\mathrm{d}s(x), \quad \varphi,\psi\in H_{00}^{\frac12}(\Gamma_{ij}).\end{aligned}$$

文献[28]得到了下列结果.

定理 11 存在两个正数 α、β 使得

$$\alpha b_h(w,w)\leqslant a_h(w,w)\leqslant \beta b_h(w,w), \quad \forall w\in \mathring{V}_h,$$

而数 α、β 满足 $\beta/\alpha\leqslant C\left(1+\log^2\dfrac{d}{h}\right)$.

文献[28]讨论了该预条件子的算法实现.

关于二阶方程非协调元的非重叠区域分解算法可参见文献[13].

以下考虑第二种类型.在前面的讨论中,都要求两个相邻子区域 Ω_i 和 Ω_j 在公共边 Γ_{ij} 上的结点重合,此时称网格是匹配的.但在某些情况下,这个要求是无法满足的.

定义 2 若两个相邻子域在其公共边上的结点不重合,则称网格是非匹配网

格.

与匹配网格情形不同的是,此时不要求每个子区域上的网格直径相同. 以 h_i 表子区域 Ω_i 上的网格直径,单元集记作 T^{h_i}. 记 $h = \max_i h_i$. 定义

$$V_{h_i} = \{ v \mid v \in C(\Omega_i), v|_{\partial\Omega} = 0, v|_e \in P_{m_i}(e), \forall e \in \Gamma^{h_i} \},$$

其中 $P_{m_i}(e)$ 表 e 上的 m_i 次多项式全体.

以 $\mathrm{T}_{ri}: H^1(\Omega_i) \to H^{\frac{1}{2}}(\partial\Omega_i)$ 表通常的迹算子. 定义

$$V_{h_i}(\partial\Omega_i) = \{ \psi \mid \psi = \mathrm{T}_{ri} v, v \in V_{h_i} \}.$$

为叙述方便,将 $H^{\frac{1}{2}}(\Gamma)$ 中函数的定义域延拓到 $\partial\Omega$ 上,其值规定为零. 设 $V(\Gamma)$ 是 $H^{\frac{1}{2}}(\Gamma)$ 的有限维子空间,记 $V(\partial\Omega_i) = V(\Gamma)|_{\partial\Omega_i}$.

假定线性算子 $\mathrm{Q}_i: V(\partial\Omega_i) \to V_{h_i}(\partial\Omega_i)$ 满足:在顶点处 $\mathrm{Q}_i\varphi_i = \varphi_i$,且有

$$\| \mathrm{Q}_i\varphi_i - \varphi_i \|_{\frac{1}{2},\partial\Omega_i} \to 0, \quad 当\ h \to 0\ 时.$$

定义

$$V_h = \{ v \mid v_i = v|_{\Omega_i} \in V_{h_i}, 存在\ \varphi \in V(\Gamma), 使\ v_i|_{\partial\Omega_i} = \mathrm{Q}_i \mathrm{I}_i \varphi \},$$

其中 $\mathrm{I}_i: V(\Gamma) \to V(\partial\Omega_i)$ 表自然的限制算子.

方程(2-1)的有限元逼近解 $u_h \in V_h$ 满足

$$\sum_{i=1}^{N} a_i(u_h, v_h) = \sum_{i=1}^{N} (f, v_h)_{\Omega_i}, \quad \forall v_h \in V_h, \tag{2-29}$$

其中

$$a_i(u,v) = \int_{\Omega_i} \Delta u \cdot \Delta v \mathrm{d}x + \eta \int_{\Omega_i} uv \mathrm{d}x, \quad u, v \in H^1(\Omega_i),$$

$$(f, v)_{\Omega_i} = \int_{\Omega_i} fv \mathrm{d}x, \quad v \in L^2(\Omega).$$

对 $\varphi_i \in V_{h_i}(\partial\Omega_i)$,以 $\mathrm{T}_{ri}^{-1}\varphi_i \in V_{h_i}$ 表 φ_i 在 Ω_i 上的离散 a_i 扩张函数.

定义离散斯德罗夫-庞加莱算子 $\mathrm{S}_i: V(\partial\Omega_i) \to V(\partial\Omega_i)$ 为

$$\langle \mathrm{S}_i\varphi_i, \psi_i \rangle = a_i(\mathrm{T}_{ri}^{-1}\mathrm{Q}_i\varphi_i, \mathrm{T}_{ri}^{-1}\mathrm{Q}_i\psi_i), \quad \varphi_i \in V(\partial\Omega_i), \quad \forall \psi_i \in V(\partial\Omega_i),$$

其中 $<\cdot,\cdot>$ 表 $\partial\Omega_i$(或 Γ)上的 L^2 内积.

定义 $\varphi \in V(\Gamma)$,使得 $\mathrm{Q}_i\mathrm{I}_i\varphi = \mathrm{T}_{ri} u_{h_i}$,其中 $u_{h_i} = u_h|_{\Omega_i} (i = 1, \cdots, N)$.

由(2-29)式和 S_i 的定义可推得关于 φ 的界面方程为(参见文献[14]和[31]):

$$\left(\sum_{i=1}^{N} \mathrm{I}_i^{\mathrm{T}} \mathrm{S}_i \mathrm{I}_i \right) \varphi = - \sum_{i=1}^{N} \mathrm{I}_i^{\mathrm{T}} \mathrm{Q}_i^{\mathrm{T}} \mathrm{T}_{ri}^{-\mathrm{T}} f, \quad \varphi \in V(\Gamma), \tag{2-30}$$

其中 $\mathrm{Q}_i^{\mathrm{T}}: V_{h_i}(\partial\Omega_i) \to V(\partial\Omega_i)$ 表 Q_i 的共轭算子;$\mathrm{I}_i^{\mathrm{T}}$ 和 $\mathrm{T}_{ri}^{-\mathrm{T}}$ 的意义同第2节.

注 2.13: 由于网格是非匹配的,故不能要求 $V_{h_i}(\partial\Omega_i) = V(\partial\Omega_i)$,因此不得不引进算子 $\mathrm{Q}_i \neq \mathrm{I}$. 易知,此时有 $V_h \not\subset H_0^1(\Omega)$,即区域分解是非协调的.

由界面空间 $V(\Gamma)$ 的不同构造方式,Q_i 的不同取法,就得到不同的区域分解算

法.较重要的有两种：

第一种方法　简化杂交法(见文献[17]).

将 Γ_{ij} 作步长为 H 的拟一致剖分,使 Ω_i 的顶点成为分点.定义界面空间

$V(\Gamma) = \{\varphi \mid \varphi \in C(\Gamma), \varphi$ 在每个 Γ_{ij} 上是分段 n 次多项式$\}$,

并将 Q_i 取作通常的结点插值算子.

文献[17]在条件 $\lim\limits_{h\to 0} h/H = 0$ 和 $m_i < n$ 下证明了相应有限元解 u_h 的存在唯一性,并给出了误差估计.

第二种方法　粘接元(Mortar element)法(见文献[1]).

设想每个 Γ_{ij} 有两侧(不同侧有不同的网格点),任意取定第 k 条边(按某种方式编号)的一侧,记作 γ_k,它们满足 $\bigcup\limits_k \gamma_k = \Gamma$.将 γ_k 称为粘接(mortar)边,而另一侧称为非粘接边,记作 γ_k'.定义界面空间

$$V(\Gamma) = \{\varphi \mid \varphi \in C(\Gamma), \varphi|_{\gamma_k} \in V_h|_{\gamma_k}, \text{对每个 } k\}.$$

以 $\widetilde{V}_{h_i}(\partial\Omega_i)$ 表 $V_{h_i}(\partial\Omega_i)$ 的余维数等于顶点个数的子空间,将 Q_i 取作从 $V(\partial\Omega_i)$ 到 $\widetilde{V}_{h_i}(\partial\Omega_i)$ 的 L^2 正交投影算子.

文献[1]证明了相应有限元解 u_h 的存在唯一性,并得到了最优的误差估计.

这两种方法的界面预条件子构造与匹配网格的情形相似(参见第3节和文献[15]),但对简化杂交法,预条件界面矩阵的条件数为 $O(1 + \mathrm{lb}^2 \dfrac{d}{H})$.

注2.14: 易知,界面方程(2-30)本质上已包括了匹配网格的情形(取 $V(\partial\Omega_i) = V_{h_i}(\partial\Omega_i)$ 及 $Q_i = I$).

注2.15: 在本章的前几节,都是将界面变量取作内部变量在界面上的迹(或迹的逼近).这类区域分解法要求界面变量在内交点处连续,从而使预条件子的结构较复杂,尤其对三维问题更是如此(参见文献[3]和[20]),尽管区域分解本身对三维问题和二维问题是一样的.在第4章将介绍另一类非重叠区域分解法,将其称作**拉格朗日乘子(Lagrangian multipliers)法**,它是将界面面变量取作内部变量在界面上的外法向导数(的适当逼近).这种区域分解法克服了上述缺点,而且便于处理非匹配网格.

2.6　无界区域问题

显而易见,直接将前面几节介绍的有界区域区域分解法推广到无界区域的情形是十分困难的.本节介绍的是将有限元法与自然边界元法相耦合而对无界区域进行区域分解的方法,其算法格式属2.1节中的D-N交替法(见文献[33]).

设 Ω 为封闭曲线 Γ_0 的外部区域,考虑 Ω 上的泊松方程的狄利克雷问题:求 u 使

$$\begin{cases} -\Delta u = f, & \text{在 } \Omega \text{ 内}, \\ u = g, & \text{在 } \Gamma_0 \text{ 上}. \end{cases} \tag{2-31}$$

附加 u 有界这一无穷远边界条件，它有唯一解.

作半径为 R 的圆周 Γ 包围 Γ_0，使 $\mathrm{dist}(\Gamma,\Gamma_0)>0$. 人工边界 Γ 将 Ω 分成内子区域 Ω_1 和外子区域 Ω_2.

不失一般性，将问题(2-31)中的 g 取作 0. 根据[34]，该边值问题等价于下列有限元与自然边界元相耦合的变分问题：求 $u\in \mathring{H}^1(\Omega_1)$，使得

$$D_1(u,v)+\widehat{D}_2(u,v)=\iint_{\Omega_1} fv\mathrm{d}x,\quad \forall v\in \mathring{H}^1(\Omega_1),\tag{2-32}$$

其中

$$\mathring{H}^1(\Omega_1)=\{v\mid v\in H^1(\Omega_1),v|_{\Gamma_0}=0\},$$

$$D_1(u,v)=\iint_{\Omega_1}\nabla u\cdot\nabla v\mathrm{d}x,$$

$$\widehat{D}_2(u,v)=\int_{\Gamma} v\mathrm{S}_2u\mathrm{d}s+\int_{\Gamma} v\left[\iint_{\Omega_2}G_h^{(2)}f\mathrm{d}x\right]\mathrm{d}s,$$

而 S_2 是对应于 Ω_2 的斯德罗夫－庞加莱算子（在文献[34]中称为自然积分算子），$G^{(2)}$ 为算子 Δ 在 Ω_2 上的格林函数. 由于 Γ 是圆，故可求得

$$(\mathrm{S}_2u)(R,\theta)=-\frac{1}{4\pi R}\int_0^{2\pi}\frac{u(R,\theta')}{\sin^2(\theta-\theta')/2}\mathrm{d}\theta',\quad 0\leqslant\theta\leqslant 2\pi,$$

$$-G_h^{(2)}(R,\theta;r,\theta')=\frac{r^2-R^2}{2\pi[R^2+r^2-2Rr\cos(\theta-\theta')]},\quad R\leqslant r<+\infty.$$

将人工边界 Γ 作 m 等分，同时在 Ω_1 内作通常的三角形或四边形有限元剖分，使 Γ 上的等分点与 Γ 上的有限元结点重合. 设 $V_h\subset \mathring{H}^1(\Omega_1)$ 为相应的有限元函数空间，则相应于方程(2-32)的离散问题是：求 $u_h\in V_h$，使

$$D_1(u_h,v_h)+\widehat{D}_2(u_h,v_h)=\iint_{\Omega_1} fv_h\mathrm{d}x,\quad \forall v_h\in V_h.\tag{2-33}$$

由此可得线性系统（取节点基函数）

$$\begin{bmatrix}A_{11} & A_{12}\\ A_{12}^{\mathrm{T}} & A_{22}+S_h^{(2)}\end{bmatrix}\begin{bmatrix}X_1\\ X_2\end{bmatrix}=\begin{bmatrix}b_1\\ b_2\end{bmatrix},\tag{2-34}$$

其中 X_1 和 X_2 分别为 Ω_1 内部的节点坐标向量和 Γ 上的节点坐标向量；$A=\begin{bmatrix}A_{11} & A_{12}\\ A_{12}^{\mathrm{T}} & A_{22}\end{bmatrix}$ 为 Ω_1 上的有限元刚度矩阵，$S_h^{(2)}$ 为 S_2 的刚度矩阵（对应于 Γ 上的结点基）.

传统的耦合法是直接求解(2-34)式，但由于 $S_h^{(2)}$ 的出现，它的系数矩阵失去了标准有限元刚度矩阵的带状稀疏结构.

现利用 D-N 交替法将 $S_h^{(2)}$ 分离出来. 将(2-34)式写成如下形式：

$$\begin{bmatrix}A_{11} & A_{12}\\ A_{12}^{\mathrm{T}} & A_{22}\end{bmatrix}\begin{bmatrix}X_1\\ X_2\end{bmatrix}=\begin{bmatrix}b_1\\ b_2-S_h^{(2)}X_2\end{bmatrix}.$$

再作迭代($n = 0,1,\cdots$)

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{12}^{\mathrm{T}} & A_{22} \end{bmatrix} \begin{bmatrix} X_1^n \\ X_2^n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - S_h^{(2)}\Lambda_n \end{bmatrix}, \tag{2-35}$$

及

$$\Lambda_{n+1} = \theta_n X_2^n + (1-\theta_n)\Lambda_n, \quad 0 < \theta_n < 1. \tag{2-36}$$

易知,当 Λ_n 给定时,求 $S_h^{(2)}\Lambda_n$ 相当于在 Ω_2 上解狄利克雷问题,而求解(2-35)式相当于在 Ω_1 上解诺伊曼问题.因此,(2-35)和(2-36)式相当于对 Ω 作两子域区域分解时的 D-N 交替法.

文献[33]得到了下列收敛性结果.

定理 12　对任意的松弛因子 $\theta_n \in (0,1)$,上述迭代算法均收敛,且当 $\theta_n = 2/3$ 时收敛得最快.此时,压缩因子为 1/3.

3　重叠区域分解法

重叠区域分解法又称**施瓦兹交替法**,其思想源于德国数学家施瓦兹(H.A. Schwarz)100年前的理论工作(见文献[27]).设想方程的定义区域是很不规则的,但它可看成两个相互重叠的规则子区域的并集.他用交替方法证明了,如果方程在这两个子区域上存在解,则原问题的解也存在.该方法的本质是将复杂区域上的问题化为简单区域上的问题考虑.将他的思想用于数值计算是40年后的事,早期文献见[16],[25]和[26].

本章考虑简单模型

$$\begin{cases} -\Delta u = f, & \text{在 } \Omega \text{ 内}, \\ u = 0, & \text{在 } \partial\Omega \text{ 上}, \end{cases} \tag{3-1}$$

其中 $\Omega \subset \mathbf{R}^2$ 是一多边形区域,$f \in L^2(\Omega)$.

为易于理解,仍先考虑两子域情形.

3.1　两子域情形

将 Ω 分解成两个相互重叠的子区域 $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$, $\Omega_1 \cap \Omega_2 = \Omega_0 \neq \varnothing$.设 Γ_1 和 Γ_2 分别是 Ω_1 的边界和 Ω_2 的边界位于 Ω 内的部分.

对模型(3-1),经典施瓦兹交替法可描述为:取定初值 $u_2^0 \in H^1(\Omega_2)$,对 $n = 1, 2\cdots$ 作迭代

步1　在 Ω_1 上解狄利克雷边值问题

$$\begin{cases} -\Delta u_1^n = f, & \text{在 } \Omega_1 \text{ 内}, \\ u_1^n = 0, & \text{在 } \partial\Omega_1 \setminus \Gamma_1 \text{ 上}, \\ u_1^n = u_2^{n-1}|_{\Gamma_1}, & \text{在 } \Gamma_1 \text{ 上}. \end{cases}$$

得到 u_1^n.

步 2　在 Ω_2 上解狄利克雷边值问题

$$\begin{cases} -\Delta u_2^n = f, & 在\ \Omega_2\ 内, \\ u_2^n = 0, & 在\ \partial\Omega_2 \setminus \Gamma_2\ 上, \\ u_2^n = u_1^n|_{\Gamma_2}, & 在\ \Gamma_2\ 上. \end{cases}$$

文献[27]证明了 u_1^n 和 u_2^n 均收敛.将其极限分别记作 u_1 和 u_2,由此得到原问题的解 u:$u|_{\Omega_i} = u_i$($i = 1,2$,自然,在 Ω_0 上有 $u_1 = u_2$).

现将这种方法用于数值计算.

定义双线性型

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}x, \quad u,v \in H^1(\Omega),$$

$$a_i(u,v) = \int_{\Omega_i} \nabla u \cdot \nabla v \mathrm{d}x, \quad u,v \in H^1(\Omega_i), i = 1,2.$$

方程(3-1)的变分问题是:求 $u \in H_0^1(\Omega)$,使

$$a(u,v) = (f,v), \forall v \in H_0^1(\Omega). \tag{3-2}$$

将 Ω_1 和 Ω_2 作直径为 h 的拟一致、正规三角形或四边形单元剖分,使其在 Γ_1、Γ_2 及 Ω_0 上的结点重合.显然,它们的全体构成 Ω 的有限元剖分.

沿用上一章的空间记号.

问题(3-2)的有限元问题是:求 $u_h \in V_h$,使

$$a(u_h,v_h) = (f,v_h), \quad \forall v_h \in V_h. \tag{3-3}$$

由经典施瓦兹交替法,可得到相应的离散算法如下(略有改动):

算法 1　乘性施瓦兹交替法

给定初值 $u_h^0 \in V_h$,对 $n = 1,2\cdots$ 作迭代

步 1　求 $\tilde{u}_{h1}^n \in V_h^1$,使

$$\begin{cases} a_1(\tilde{u}_{h1}^n, V_h) = \int_{\Omega_1} f v_h \mathrm{d}x, & \forall v_h \in \mathring{V}_h^1, \\ \tilde{u}_{h1}^n = u_h^{n-1}|_{\Gamma_1}, & 在\ \Gamma_1\ 上; \end{cases}$$

定义

$$u_{h1}^n = \begin{cases} \tilde{u}_{h1}^n, & 在\ \Omega_1\ 上, \\ u_h^{n-1}, & 在\ \Omega \setminus \Omega_1\ 上. \end{cases}$$

步 2　求 $\tilde{u}_{h2}^n \in V_h^2$,使

$$\begin{cases} a_2(\tilde{u}_{h2}^n, v_h) = \int_{\Omega_2} f v_h \mathrm{d}x, & \forall V_h \in \mathring{V}_h^2, \\ \tilde{u}_{h2}^n = u_{h1}^n|_{\Gamma_2}, & 在\ \Gamma_2\ 上; \end{cases}$$

定义

$$u_h^n = \begin{cases} u_{h1}^n, & \text{在 } \Omega_1 \text{ 上}, \\ \tilde{u}_{h2}^n, & \text{在 } \Omega_2 \text{ 上}. \end{cases}$$

算法1是串行的，故习惯上称为乘性施瓦兹交替法（或乘性施瓦兹算法）. 自然，也可构造并行的交替法.

算法2 加性施瓦兹交替法

取定初值 $u_h^0 \in V_h$，对 $n = 1,2,\cdots$ 作迭代

步1 并行求解 $u_{hi}^n \in \mathring{V}_h^i$，使

$$\begin{cases} a_i(u_{hi}^n, v_h) = \int_{\Omega_i} f v_h \mathrm{d}x, & \forall v_h \in \mathring{V}_h^i, \\ u_{hi}^n = u_h^{n-1}|_{\Gamma_i}, & \text{在 } \Gamma_i \text{ 上} \end{cases} \quad (i = 1,2).$$

步2 延拓 u_{hi}^n 至整个 Ω 上，

$$\tilde{u}_{hi}^n = \begin{cases} u_{hi}^n, & \text{在 } \Omega_i \text{ 上}, \\ u_h^{n-1}, & \text{在 } \Omega \setminus \Omega_i \text{ 上} \end{cases} \quad (i = 1,2),$$

并计算

$$u_h^n = \frac{1}{2}(\tilde{u}_{h1}^n + \tilde{u}_{h2}^n).$$

以下给出施瓦兹算法的投影解释和代数解释.

以 $\mathrm{P}_i: V_h \to \mathring{V}_h^i$ 表相对于内积 $a(\cdot,\cdot)$ 的正交投影算子. 记 $e_n = u_h - u_h^n$. 易推得（参见文献[20]和[29]）：

对算法1有

$$u_h^n = u_h^{n-1} + (\mathrm{P}_1 + \mathrm{P}_2 - \mathrm{P}_2\mathrm{P}_1)e_{n-1}; \tag{3-4}$$

对算法2有

$$u_h^n = u_h^{n-1} + \frac{1}{2}(\mathrm{P}_1 + \mathrm{P}_2)e_{n-1}. \tag{3-5}$$

值得指出的是，虽然 e_{n-1} 无法求出，但 $\mathrm{P}_1 e_{n-1}$ 和 $\mathrm{P}_2 e_{n-1}$ 可算出. 因此，从(3-4)和(3-5)式知，算法1和算法2本质上是预条件里查德森迭代. 这可从即将给出的代数形式更清楚地看出.

对 V_h 的结点基函数，设 u_h 的坐标向量为 $\boldsymbol{x}$. 以 $\boldsymbol{A}$ 表 $a(\cdot,\cdot)$ 对应的刚度矩阵，$\boldsymbol{b}$ 表方程(3-3)的右端向量. 则(3-3)式的代数形式为

$$\boldsymbol{Ax} = \boldsymbol{b}.$$

将 $\boldsymbol{A}$ 按 Ω_1 内的结点基函数和 $\Omega \setminus \Omega_1$ 上的基函数进行分块，把由 Ω_1 内结点基函数构成的子块（方阵）记作 $\boldsymbol{A}_{11}$. 记

$$\boldsymbol{B}_1 = \begin{bmatrix} \boldsymbol{A}_{11}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

同理，将 $\boldsymbol{A}$ 按 Ω_2 内的结点基函数和 $\Omega \setminus \Omega_2$ 上的基函数进行分块. 把由 Ω_2 内结点基函数构成的子块记作 $\boldsymbol{A}_{22}$. 记

$$B_2 = \begin{bmatrix} 0 & 0 \\ 0 & A_{22}^{-1} \end{bmatrix}.$$

易知,(3-4)和(3-5)式的代数形式分别为(参见文献[29])

算法1：　$x_n = x_{n-1} + (B_1 + B_2 - B_2 A B_1)(b - A x_{n-1}).$　(3-6)

算法2：　$x_n = x_{n-1} + \frac{1}{2}(B_1 + B_2)(b - A x_{n-1}).$　(3-7)

其中 x_n 表 u_h^n 的坐标向量.

由(3-6)和(3-7)式可知,算法1和算法2本质上是预条件子里查得森迭代,预条件子分别为 $B_1 + B_2 - B_2AB_1$ 和 $\frac{1}{2}(B_1 + B_2)$.前者像块高斯-塞德尔迭代,后者像块雅可比迭代.

由(3-4)和(3-5)式,按标准的方法可得到算法1和算法2的收敛性结果(参见文献[29]和[9]).

定理1　设 Ω_1 和 Ω_2 的"尺寸"均为 d,Ω_0 的"尺寸"为 δ,则存在与 h、d 和 δ 均无关的常数 C,使对算法1和算法2,均有

$$\| e_n \| \leqslant \left(1 - \frac{1}{C(1 + d/\delta)}\right)^{\frac{1}{2}} \| e_{n-1} \|, \tag{3-8}$$

其中 $\|\cdot\|$ 表由 $a(\cdot,\cdot)$ 生成的范数.

注3.1：定理1表明,算法1和算法2总是收敛的,但收敛速度依赖于 d/δ.一般说来,当重叠部分 Ω_0 越小时,收敛得越慢.反之,要提高收敛速度,必须增大重叠区域,即必须以增加计算量为代价.

3.2　组合网格方法

组合网格方法是处理奇异性问题的一种新技术,尽管它形式上无通常的区域分解,但它可归结为施瓦兹交替法的框架.这种方法最早在文献[22]中研究.

将 Ω 作直径为 H 的拟一致、正规三角形或四边形单元剖分,相应的线性或双线性有限元空间记作 V_H(边界条件与原问题相同).设想问题(3-1)的解析解在某点 $x_0 \in \Omega$ 处有奇异性.取一含有 x_0 的某子区域 Ω_0,使 Ω_0 恰为若干个单元的并集,且 Ω_0 的"尺寸"与 Ω 的"尺寸"保持一定的比例(当 $H \to 0$ 时).现对 Ω_0 中的单元作适当加密,使新单元的直径为 $h \ll H$.将 Ω_0 上新的线性或双线性有限元空间(取零边值,并零延拓至 Ω 上)记作 $\mathring{V}_h$.令 $V_{h,H} = V_H + \mathring{V}_h$.通常将 $V_{h,H}$ 称为**组合网格有限元空间**.

对应于此空间,方程(3-2)的有限元问题是:求 $u_{h,H} \in V_{h,H}$,使

$$a(u_{h,H}, v) = (f, v), \quad \forall v \in V_{h,H}. \tag{3-9}$$

如直接求解(3-9)式,显然不方便,因为其刚度矩阵不具有标准有限元刚度矩阵的带状稀疏结构,且阶数较高(通常的局部加密方法也有同样问题).以下采用施瓦兹交替法求解,可使粗网格方程和细网格方程独立求解.

算法3　组合网格交替法

取定初值 $u_{h,H}^0 \in V_{h,H}$. 设已求得 $u_{h,H}^n \in V_{h,H}$,则按下列方式求 $u_{h,H}^{n+1}$:

步1　求 $u_h \in \mathring{V}_h$,使

$$a(u_h, v_h) = (f, v_h) - a(u_{h,H}^n, v_h), \quad \forall v_h \in \mathring{V}_h,$$

并令 $u_{h,H}^{n+1/2} = u_{h,H}^n + u_h$.

步2　求 $u_H \in V_H$,使

$$a(u_H, v_H) = (f, v_H) - a(u_{h,H}^{n+1/2}, v_H), \quad \forall v_H \in \mathring{V}_H,$$

并令 $u_{h,H}^{n+1} = u_{h,H}^{n+1/2} + u_H$.

关于该算法有下列收敛性结果.

定理2　设 $e_n = u_{h,H} - u_{h,H}^n$ 是算法3对应的误差函数,则存在与 h、H 无关的常数 $\rho < 1$,使

$$\| e_{n+1} \| \leqslant \rho \| e_n \|. \tag{3-10}$$

因此,算法3恒收敛,且收敛速度不随网格"尺寸"变小而明显减慢.

文献[23]中的数值实验表明,算法3收敛得很快,即压缩因子 ρ 很小.因此,原文将这种方法称为快速自适应组合网格法(fast adaptive composite grid methods).

定理2的证明可在文献[23]中找到,以下给出更为简洁的证明.

以 $P_h: V_{h,H} \to \mathring{V}_h$ 和 $P_H: V_{h,H} \to V_H$ 表在内积 $a(\cdot,\cdot)$ 下的正交投影算子,易推得误差传播关系为(参见文献[20]或[23])

$$e_{n+1} = (I - P_H)(I - P_h)e_n. \tag{3-11}$$

故只须估计算子 $(I - P_H)(I - P_h)$ 的范数.

先证明:存在常数 C,使对任意 $v \in \overline{V}_{h,H}$,均有 $v_1 \in V_H$,满足 $v - v_1 \in \mathring{V}_h$,和

$$\| v_1 \|^2 \leqslant C \| v \|^2 \quad (C \geqslant 1). \tag{3-12}$$

构造函数 $v_H \in V_H|_{\Omega_0}$,使 $v_H|_{\partial\Omega_0} = v|_{\partial\Omega_0}$,而在 Ω_0 内部是离散调和的,即 v_H 是 $v|_{\partial\Omega_0}$ 在 Ω_0 上的离散调和扩张函数.故存在常数 C_0,使

$$\| v_H \|_{\Omega_0}^2 \leqslant C_0 | v_H |_{\frac{1}{2},\partial\Omega_0}^2 = C_0 | v |_{\frac{1}{2},\partial\Omega_0}^2. \tag{3-13}$$

定义 $v_1 \in V_H$ 如下:

$$v_1 = \begin{cases} v_H, & \text{在 } \Omega_0 \text{ 上}, \\ v, & \text{在 } \Omega \setminus \Omega_0 \text{ 上}. \end{cases}$$

由(3-13)式和迹定理有

$$\begin{aligned} \| v_1 \|^2 &= \| v_1 \|_{\Omega_0}^2 + \| v_1 \|_{\Omega\setminus\Omega_0}^2 = \| v_H \|_{\Omega_0}^2 + \| v \|_{\Omega\setminus\Omega_0}^2 \\ &\leqslant C_0 | v |_{\frac{1}{2},\partial\Omega_0}^2 + \| v \|_{\Omega\setminus\Omega_0}^2 \leqslant C \| v \|_{\Omega_0}^2 + \| v \|_{\Omega\setminus\Omega_0}^2 = C \| v \|^2. \end{aligned}$$

由 $\mathring{V}_h$ 和 v_1 的定义易知 $v - v_1 \in \mathring{V}_h$,即(3-12)式成立.

现取 $v = (I - P_h)e_n$,则存在分解 $(I - P_h)e_n = v_1 + v_2$,其中 $v_1 \in V_H$, $v_2 \in \mathring{V}_h$,满足

$$\| v_1 \|^2 \leqslant C \| (I - P_h)e_n \|^2.$$

故由柯西不等式得

$$\begin{aligned}\|(\mathrm{I}-\mathrm{P}_h)e_n\|^2 &= a((\mathrm{I}-\mathrm{P}_h)e_n, v_1+v_2)\\ &= a((\mathrm{I}-\mathrm{P}_h)e_n, v_1) = a(\mathrm{P}_H(\mathrm{I}-\mathrm{P}_h)e_n, v_1)\\ &\leqslant \|\mathrm{P}_H(\mathrm{I}-\mathrm{P}_h)e_n\|\cdot\|v_1\|\\ &\leqslant \sqrt{C}\,\|\mathrm{P}_H(\mathrm{I}-\mathrm{P}_h)e_n\|\cdot\|(\mathrm{I}-\mathrm{P}_h)e_n\|,\end{aligned}$$

即

$$\|(\mathrm{I}-\mathrm{P}_h)e_n\|^2 \leqslant C\|\mathrm{P}_H(\mathrm{I}-\mathrm{P}_h)e_n\|^2.$$

由此及(3-11)式立得(3-10)式,其中 $\rho=(1-\frac{1}{C})^{\frac{1}{2}}$.

算法3是串行算法,自然也可讨论相应的并行算法.

算法4　组合网格的加性施瓦兹交替法

取定初值 $u_{h,H}^0\in V_{h,H}$. 若已求得 $u_{h,H}^n\in V_{h,H}$,则按下列方式求 $u_{h,H}^{n+1}$:

步1　并行计算 $u_H\in V_H$, $u_h\in\mathring{V}_h$ 和 $w_H\in V_H\cap\mathring{V}_h$,分别满足

$$a(u_{h,H}^n-u_H, v_H)=(f,v_H),\quad \forall v_H\in V_H,$$

$$a(u_{h,H}^n-u_h, v_h)=(f,v_h),\quad \forall v_h\in\mathring{V}_h,$$

$$a(u_{h,H}^n-w_H, v_H)=(f,v_H),\quad \forall v_H\in V_H\cap\mathring{V}_h.$$

步2　计算 $u_{h,H}^{n+1}=u_{h,H}^n-(u_H+u_h-w_H)$.

该算法与算法3有相似的收敛性结果(参见文献[20]).

定理3　对算法4,存在与网格"尺寸"h、H无关的压缩因子 $\rho<1$,使

$$\|e_{n+1}\|\leqslant\rho\|e_n\|. \tag{3-14}$$

注3.2: 对组合网格方法有各种不同的迭代变形(参见文献[20]). 此外,可将其思想推广,即不限于有限元空间的组合.

注3.3: 对上节中的两子域重叠区域分解,显然有 $V_h=\mathring{V}_h^1+\mathring{V}_h^2$. 因此,本节的组合网格方法在理论上与上节的区域分解方法等价(重叠部分为 $V_H\cap\mathring{V}_h$),本质差异是,本节中有 $\Omega_0\subset\Omega$.

3.3　多子域情形

多子域重叠区域分解的思想同两子域情形,但问题复杂得多.

首先,对区域 Ω 作初始正规三角形或四边形剖分 $\bar{\Omega}=\bigcup_{i=1}^m\bar{\Omega}_i$,其中 $\Omega_i\cap\Omega_j=\varnothing$,它们的直径均为 H;其次,对每个 Ω_i 作直径为 h 的拟一致、正规三角形或四边形剖分,使在公共边上的结点重合,它们的全体构成 Ω 的拟一致剖分(直径为 h);最后,对每个 Ω_i 构造真包含 Ω_i 的子域 Ω_i',使 Ω_i' 的顶点是上述剖分的结点,且 $\partial\Omega_i$ 不通过任何小单元. 以 $V_h\subset H_0^1(\Omega)$ 表 Ω 上的分片线性或双线性有限元空间,V_H 表示对应于初始剖分的线性有限元空间. 记 $\mathring{V}_h^i=V_h\cap H_0^1(\Omega_i')(i=1,\cdots,m)$. 设

$$\mathrm{dist}(\partial\Omega_i'/\partial\Omega, \partial\Omega_i\setminus\partial\Omega)=\delta\geqslant h,\quad i=1,\cdots,m.$$

显然，$\overline{\Omega} = \bigcup_{i=1}^{m} \overline{\Omega_i'}$ 是一特殊的多子域重叠区域分解.

先给出两种基本算法.

算法 5 多子域乘性施瓦兹交替法

取定初值 $u_h^0 \in V_h$，若已求得 $u_h^n \in V_h$，则按下列方式求 u_h^{n+1}：

步 1 求 $u_H \in V_H$，使

$$a_0(u_H, v_H) = (f, v_H) - a(u_h^n, v_H), \quad \forall v_H \in V_H,$$

并令 $u_h^{n+\frac{1}{m+1}} = u_h^n + u_H$.

步 2 求 $u_{h1} \in \mathring{V}_h^1$，使

$$a_1(u_{h1}, v_h) = (f, v_h) - a(u_h^{n+\frac{1}{m+1}}, v_h), \quad \forall v_h \in \mathring{V}_h^1,$$

并令 $u_h^{n+\frac{2}{m+1}} = u_h^{n+\frac{1}{m+1}} + u_{h1}$.

……

步 $m+1$ 求 $u_{hm} \in \mathring{V}_h^m$，使

$$a_m(u_{hm}, v_h) = (f, v_h) - a(u_h^{n+\frac{m}{m+1}}, v_h), \quad \forall v_h \in \mathring{V}_h^m,$$

并令 $u_h^{n+1} = u_h^{n+\frac{m}{m+1}} + u_{hm}$.

算法 6 多子域加性施瓦兹交替法

取初始值 $u_h^0 \in V_h$，若已求得 $u_h^n \in V_h$，则按下列方式求 u_h^{n+1}：

步 1 并行求解 $u_{hi}^n \in H^1(\Omega_i') \cap V_h (i = 1, \cdots, m)$，使

$$\begin{cases} a_i(u_{hi}^{n+1}, v_h) = (f, v_h), & \forall v_h \in \mathring{V}_h^i, \\ u_{hi}^{n+1} = u_h^n |_{\partial\Omega_i'}, & \text{在 } \partial\Omega_i' \text{ 上}. \end{cases}$$

步 2 求 $u_{h_0}^{n+1} \in V_H$，满足

$$a_0(u_{h_0}^{n+1}, v_H) = (f, v_H), \quad \forall v_H \in V_H.$$

步 3 令 $u_h^{n+1} = \dfrac{1}{m+1}\sum_{i=0}^{m} u_{hi}^{n+1}$.

注 3.4：若不是零边值问题，则上述算法略有变化(参见文献[20]).

以 $P_i: V_h \to \mathring{V}_h^i (i = 1, \cdots, m)$ 和 $P_0: V_h \to V_H$ 表示相对于内积 $a(\cdot, \cdot)$ 的正交投影算子.易推得上述算法的误差传播关系为(参见文献[20])

算法 5：
$$e_{n+1} = (I - P_m)(I - P_{m-1})\cdots(I - P_0)e_n, \tag{3-15}$$

算法 6：
$$e_{n+1} = \left[I - \frac{1}{m+1}(P_0 + P_1 + \cdots + P_m)\right]e_n. \tag{3-16}$$

由此可得这两种算法的代数形式为(所有符号均按本章第1节的方式定义)

算法 5：

$$x_{n+\frac{1}{m+1}} = x_n + B_H(b - Ax_n),$$

$$x_{n+\frac{2}{m+1}} = x_{n+\frac{1}{m+1}} + B_1(b - Ax_{n+\frac{1}{m+1}}),$$

……

$$x_{n+1} = x_{n+\frac{m}{m+1}} + B_m(b - Ax_{n+\frac{m}{m+1}}).$$

算法 6:

$$x_{n+1} = x_n + \frac{1}{m+1} \quad (B_0 + B_1 + \cdots + B_m)(b - Ax_n). \tag{3-17}$$

为得到这两种算法的收敛性结果,需要下列估计式(详见文献[9])

定理4　存在与 h、H、δ 无关的常数 C,使对任何 $v \in V_h$,均有分解 $v = \sum_{i=0}^{m} v_i$,其中 $v_0 \in V_h, v_i \in \mathring{V}_h^i (i = 1, \cdots, m)$,满足

$$\sum_{i=0}^{m} \| v_i \|^2 \leqslant C(1 + H/\delta) \| v \|^2. \tag{3-18}$$

由此,按标准的方式可证明(参见文献[20]):

定理5　算法 5 和算法 6 均收敛,且有

算法 5:　$\| e_{n+1} \| \leqslant \left(1 - \frac{1}{C(1 + H/\delta)}\right)^{\frac{1}{2}} \| e_n \|$;

算法 6:　$\| e_{n+1} \| \leqslant \left(1 - \frac{1}{Cm(1 + H/\delta)}\right) \| e_n \|$.

注 3.5: 从定理 5 可知,加性算法一般比乘性算法慢,特别是当 m 较大时.但是多子域的乘性算法结构较复杂,它在区域分解法中很少使用(它主要用于多层组合网格法和多层网格法中).

以下介绍算法 6 与共轭梯度法的结合,这样可大大提高收敛速度.

将(3-16)式写成

$$u_h^{n+1} = u_h^n + \frac{1}{m+1}\sum_{i=0}^{m} \mathrm{P}_i(u_h - u_h^n),$$

若记 $\mathrm{P} = \sum_{i=0}^{m} \mathrm{P}_i, g = \mathrm{P}u_h$,则上式即为

$$u_h^{n+1} = u_h^n + \frac{1}{m+1}(g - \mathrm{P}u_h^n).$$

因此算法 6 本质上就是对下列方程作里查德森迭代(松弛因子为$\frac{1}{m+1}$,注意尽管 u_h 不能得到,但 $g = \mathrm{P}u_h$ 可以求出)

$$\mathrm{P}u_h = g. \tag{3-19}$$

从代数上看,算法 6 相当于以 $B = \sum_{i=0}^{m} B_i$ 为预条件子,对原代数系统

$$Ax = b \tag{3-20}$$

作预条件的里查德森迭代(注意(3-17)式).

基于这种观察,自然想到对(3-19)式作共轭梯度迭代或对(3-20)式作预条件共轭梯度迭代.在每个迭代步只涉及子空间 V_H 和 $\mathring{V}_h^i$ 上的求解,且它们是并行的.这种迭代的收敛速度取决于算子 P 的条件数.由(3-18)式可得到下列结果.

定理6　算子 P 的条件数

$$\mathrm{cond}(\mathrm{P}) \leqslant C(1 + H/\delta), \tag{3-21}$$

其中常数 C 与网格参数 h、H 和 δ 无关.

在本章前面介绍的算法中,局部可解子均是精确的.事实上,对非精确的局部可解子,结果并无本质差别,这是重叠区域分解法的优点.

设 $b_0(u,v)$ 和 $b_i(u,v)(i=1,\cdots,m)$ 分别为定义在 $V_H \times V_H$ 和 $\mathring{V}_h^i \times \mathring{V}_h^i$ 上的双线性型.假定存在常数 ω,使

$$a(u,u) \leqslant \omega b_i(u,u), \quad \forall u \in \mathring{V}_h^i(\text{或 } V_H).$$

定义投影型算子 $\mathrm{T}_i: V_h \to \mathring{V}_h^i$(或 V_H)如下

$$b_i(\mathrm{T}_i u, v) = a(u,v), \quad \forall v \in \mathring{V}_h^i(\text{或 } V_H).$$

定义1　若将算法5和算法6中的双线性型 $a_i(\cdot,\cdot)$ 由 $b_i(\cdot,\cdot)$ 代替,即投影算子 P_i 由 T_i 代替,则称 T_i 为 $\mathring{V}_h^i$(或 V_H)上的**非精确可解子**(设 $b_i(\cdot,\cdot) \neq a_i(\cdot,\cdot)$).

定义2　称矩阵 $\boldsymbol{\varepsilon} = [\varepsilon_{ij}]$ 为加强的柯西-施瓦兹常数矩阵,若 ε_{ij} 是满足下列不等式的最小常数

$$|a(v_i,v_j)| \leqslant \varepsilon_{ij} \|v_i\| \cdot \|v_j\|, \quad \forall v_i \in \mathring{V}_h^i, v_j \in \mathring{V}_h^j; i,j \geqslant 1.$$

定义数 C_0,使对任意 $v \in V_h$ 存在分解 $v = \sum_{i=0}^{m} v_i$,其中 $v_0 \in V_H$,$v_i \in \mathring{V}_h^i(i=1,\cdots,m)$,满足

$$\sum_{i=0}^{m} b_i(v_i,v_i) \leqslant C_0^2 \|v\|.$$

记 $\mathrm{T} = \sum_{i=0}^{m} T_i$.将(3-19)式中的P由T代替,相应共轭梯度法的收敛速度由T的条件数决定.有下列一般性结果(参见文献[29]或[9])

定理7　算子T的条件数

$$\mathrm{cond}(\mathrm{T}) \leqslant \omega(\rho(\boldsymbol{\varepsilon})+1)C_0^2, \tag{3-22}$$

其中 $\rho(\boldsymbol{\varepsilon})$ 表矩阵 $\boldsymbol{\varepsilon}$ 的谱半径.

显然,(3-22)式右端的具体值由 $b_i(\cdot,\cdot)$ 的具体形式决定。(3-21)式可看作取 $b_i(\cdot,\cdot) = a_i(\cdot,\cdot)$ 时的特例.

3.4　变分不等式问题

本节基本取材于[20],旨在以简单模型说明变分不等式的区域分解算法的特点.

考虑变分不等式问题:求 $u \in K$,使

$$a(u, v-u) \geqslant f(v-u), \quad \forall v \in K, \tag{3-23}$$

其中 $a(\cdot,\cdot)$ 是一希尔伯特空间 V 上连续、对称、强制的双线性型;$f: V \to \mathbf{R}$ 是连续线性泛函;K 是 V 的非空凸子集.

以下设 $V = H_0^1(\Omega)$，其中 $\Omega \subset \mathbf{R}^2$ 仍是多边形区域. 将 Ω 作直径为 h 的拟一致、正规三角形剖分，$V_h \subset V$ 是相应的分片线性元空间，构造 V_h 的闭凸子集 K_h，其具体形式与 K 有关，比如取 $K_h = V_h \cap K$.

(3-23) 式的有限元问题是：求 $u_h \in K_h$，使

$$a(u_h, v_h - u_h) \geqslant f(v_h - u_h), \quad \forall v_h \in K_h. \tag{3-24}$$

可以证明：当 $h \to 0$ 时，u_h 收敛于 u（参见文献[20]).

现将 Ω 作重叠型区域分解 $\Omega = \bigcup_{i=1}^{m} \Omega_i$。记 $\mathring{V}_h^i = V_h \cap H_0^1(\Omega_i)$，$K_h^i = K_h \cap \mathring{V}_h^i$，假定：$K_h = \sum_{i=1}^{m} K_h^i$.

算法 7　变分不等式的加性施瓦兹交替法

取初值 $u_h^0 \in K_h$，若已求得 $u_h^n \in K_h$，则按下列方式求 u_h^{n+1}.

步 1　并行求 $w_i^n \in K_h^i (i = 1, \cdots, m)$，使

$$a(w_i^n + u_h^n, v - w_i^n) \geqslant f(v - w_i^n), \quad \forall v \in K_h^i,$$

并令 $u_{hi}^n = u_h^n + w_i^n, i = 1, \cdots, m$；

步 2　选择 $\omega_i > 0 (i = 1, \cdots, m)$，使 $\sum_{i=1}^{m} \omega_i = 1$，并令

$$u_h^{n+1} = \sum_{i=1}^{m} \omega_i u_{hi}^n.$$

下列结果的证明见文献[20].

定理 8　若 K_h 是含原点的闭凸锥，则由算法 7 产生的序列 $\{u_h^n\}$ 收敛到问题 (3-24) 的解 u_h，且迭代误差单调下降.

为加快收敛速度，可以引进带松弛因子的算法.

算法 8　带松弛因子的加性施瓦兹算法

步 1　选定初值 $u_h^0 \in K_h$，按算法 7 计算 w_i^n.

步 2　确定松弛因子 $1 < \tau < 2$，并令

$$u_{hi}^n = u_h^n + \tau w_i^n;$$

步 3　按算法 7 步 2 的方式确定 u_h^{n+1}.

从文献[20] 中可得到下列结果.

定理 9　算法 8 比算法 7 收敛得更快。

现介绍变分不等式的一种数值解法.

算法 7 和算法 8 的步 1 归结为在 K_h^i 上求解形如(3-24) 式的变分不等式，它们等价于在 $\mathbf{R}^n$ 中的一个闭凸集 K_n 上求 $\boldsymbol{x}$，使

$$J_i(\boldsymbol{x}) = \min_{\boldsymbol{y} \in K_n} J_i(\boldsymbol{y}), \tag{3-25}$$

其中

$$J_i(\boldsymbol{y}) = \frac{1}{2}(\boldsymbol{A}_i \boldsymbol{y}, \boldsymbol{y}) - (\boldsymbol{g}, \boldsymbol{y}).$$

而 $\boldsymbol{A}_i$ 是对应于 K_h^i 的有限元刚度矩阵，$\boldsymbol{g}$ 是相应的右端项. 最常见的情形是

$$K_n = \{\boldsymbol{x} \in \mathbf{R}^n \mid x_k \leqslant C_k, 1 \leqslant k \leqslant n\},$$

其中 C_k 为固定常数(可能与 i 有关).

下列求解(3-25)式的算法参见文献[12].

算法9 求解变分不等式的投影 SOR 方法

步1 选择 $x^0 \in \mathbf{R}^n$,置 $n:=0$.

步2 对 $k=1,2,\cdots,n$ 执行

$$\widehat{x}_k = \Big(f_k - \sum_{j<k} a_{kj} x_j^{n+1} - \sum_{k<j} a_{kj} x_j^n\Big)\Big/ a_{kk},$$

$$x_k^{n+1} = x_k^n + \omega(\widehat{x}_k - x_k^n), \quad 1 \leqslant \omega < 2,$$

$$x_k^{n+1} := \min(C_k, x_k^{n+1}).$$

步3 若 $\|x^{n+1} - x^n\|_\infty \leqslant \epsilon \|x^n\|_\infty$,则停机并输出结果,否则置 $n:=n+1$ 并转步2.

这里 ϵ 是预先设置的计算精度.

3.5 非对称、非正定问题

对非对称、非正定问题,原则上仍可采用算法5和算法6.但此时为得到满意的收敛结果,需要作一系列的假定.

考虑模型问题

$$\begin{cases} \mathrm{L}u = f, & \text{在 } \Omega \text{ 内}, \\ u = 0, & \text{在 } \partial\Omega \text{ 上}, \end{cases} \tag{3-26}$$

其中 $\Omega \subset \mathbf{R}^2$ 是多角形区域,算子 L 定义为

$$\mathrm{L}u = -\sum_{i,j} \frac{\partial}{\partial x_i}\Big(a_{ij}(x)\frac{\partial u}{\partial x_j}\Big) + \sum_i b_i(x)\frac{\partial u}{\partial x_i} + c(x)u,$$

而矩阵$[a_{ij}(x)]$是对称、一致正定的,$f \in L^2(\Omega)$.

方程(3-26)的弱形式是:求 $u \in H_0^1(\Omega)$,使

$$b(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega). \tag{3-27}$$

其中

$$b(u,v) = \int_\Omega \Big(\sum_{i,j} a_{ij}(x)\frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j} + \sum_i b_i(x)\frac{\partial u}{\partial x_i}v + C(x)uv\Big)\mathrm{d}x.$$

按3.3节那样将 Ω 作重叠区域分解和有限元剖分,并沿用相应的空间记号.

对应于(3-27)式的有限元问题是:求 $u_h \in V_h$,使

$$b(u_h, v) = (f,v), \quad \forall v \in V_h. \tag{3-28}$$

以 $\mathrm{T}_0: V_h \to V_H$ 和 $\mathrm{T}_i: V_h \to \mathring{V}_h^i (i=1,\cdots,m)$ 表由双线性型 $b(\cdot,\cdot)$ 定义的"投影型"算子(假定定义的合理性)

$$b(\mathrm{T}_i u, v) = b(u,v), \quad u \in V_h, \forall v \in \mathring{V}_h^i (\text{或 } V_H).$$

用算法6求(3-28)式等价于用迭代法求解下列算子方程

$$\Big(\sum_{i=1}^m \mathrm{T}_i\Big)u_h = g. \tag{3-29}$$

由于 $T = \sum_{i=1}^{m} T_i$ 在 V_h 上不是对称正定的，故用著名的 GMRES 算法求解(3-29) 式或其对应的代数系统．其收敛速度决定于下列两个参数

$$c_p = \inf_{v \in V_h} \frac{a(Tv,v)}{a(v,v)} \text{ 和 } C_p = \sup_{v \in V_h} \frac{a(Tv,Tv)}{a(v,v)}.$$

为使以上的讨论合理，并得到 c_p 和 C_p 的满意估计，需要一系列假定．

定义双线性型

$$a(u,v) = \sum_{i,j} \int_\Omega a_{ij}(x) \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_j} \mathrm{d}x,$$

$$s(u,v) = \int_\Omega \left(\sum_i b_i(x) \frac{\partial u}{\partial x_i} v + \frac{\partial b_i(x) u}{\partial x_i} v \right) \mathrm{d}x = b(u,v) - b(v,u),$$

$$c(u,v) = \int_\Omega (c(x) - \nabla \cdot \boldsymbol{b}(x)) uv \mathrm{d}x.$$

则有

$$b(u,v) = a(u,v) + s(u,v) + c(u,v).$$

由 $a(\cdot,\cdot)$ 生成的范数记作 $\|\cdot\|_a$．假定

1° 存在常数 C，使对任何 u、$v \in H_0^1(\Omega)$ 有

$$|b(u,v)| \leqslant C \|u\|_a \cdot \|v\|_a.$$

2° 反对称项 $s(\cdot,\cdot)$ 有界，即对 u、$v \in H_0^1(\Omega)$ 有

$$|s(u,v)| \leqslant C \|u\|_a \|v\|_{L^2(\Omega)}.$$

3° 存在正常数 C，使对任何 $u \in H_0^1(\Omega)$，有

$$\|u\|_a^2 - C\|u\|_{L^2(\Omega)}^2 \leqslant b(u,u).$$

4° 对偶方程

$$b(\varphi, w) = (f, \varphi), \quad \forall \varphi \in H_0^1(\Omega)$$

的解满足

$$\|w\|_{H^{1+\theta}} \leqslant C \|f\|_{L^2(\Omega)}.$$

其中 $\theta \geqslant 1/2$．

这些假定表明，$b(\cdot,\cdot)$ 中的低阶项 $s(\cdot,\cdot)$ 和 $c(\cdot,\cdot)$ 是二阶项 $a(\cdot,\cdot)$ 的紧扰动，由此可以得到有限元分析的一些基本结果．

定理 10 若重叠部分的"尺寸"与子区域的"尺寸"保持一定的比例，则在本节假定下 c_p 有与网格"尺寸"无关的正下界，C_p 有与网格"尺寸"无关的正上界．

该定理的证明见文献[29]．它表明用 GMRES 迭代方法求解(3-29) 式时有满意的收敛速度，每个迭代步只需并行求解局部子问题，关于 GMRES 算法的细节参见文献[4] 和[29]．

4 抛物问题的拉格朗日乘子区域分解法

对椭圆问题的拉格朗日乘子区域分解法首先被文献[10] 和[18] 研究，这种区

域分解方法比传统的非重叠区域分解方法有明显的优点(参见注 2.15 和文献[14]).研究抛物问题拉格朗日乘子区域分解法的文献见[11],[14]和[30].本章基本取材于[14].

4.1　基本算法

为叙述方便,考虑简单模型

$$\begin{cases}\dfrac{\partial u}{\partial t}-\Delta u=f, & (x,t)\in\Omega\times(0,T],\\ u=0, & (x,t)\in\partial\Omega\times(0,T],\\ u(x,0)=0, & x\in\Omega,\end{cases}\tag{4-1}$$

其中 $\Omega\subset\mathbf{R}^2$ 是多边形区域;T 是一给定的正数.

将 Ω 分成 N 个多边形 Ω_i,我们先作下列假定:

H_1:所有子区域 Ω_i 的尺寸为 d,即存在常数 c_0 和 c_1(与 d 无关),使每个 Ω_i 包含(包含在)直径为 $c_0d(c_1d)$ 的圆(圆内).

H_2:对 $i\neq j$,若开边 $\Gamma'\subset\partial\Omega_i$ 和 $\Gamma''\subset\partial\Omega_j$ 有一个公共点,则 $\overline{\Gamma'}=\overline{\Gamma''}=\overline{\Omega}_i\cap\overline{\Omega}_j=\Gamma_{ij}$.以 $\Gamma=\bigcup\Gamma_{ij}\bigcup\partial\Omega$ 表整个界面.

H_3:每个 Ω_i 可分成直径为 h_i 的拟一致正规三角形或四边形单元.

对自然数 n 及 $h=\max\limits_{1\leq i\leq N}h_i$,假定

H_4:当 $h\to 0$ 时,n^2h/d 充分小.定义逼近空间如下:

以 $V_{h_i}(\Omega_i)$ 表定义在 Ω_i 上的次数不超过 m_i 的多项式空间,定义

$$V_h(\Omega)=\{v\mid v|_{\Omega_i}\in V_{h_i}(\Omega_i),i=1,2,\cdots,N\},$$

$$V_n(\Gamma_{ij})=\{\lambda_{ij}\mid\lambda_{ij}\text{ 在 }\Gamma_{ij}\text{ 上是次数不超过 }n\text{ 的多项式}\},$$

其中 $\Gamma_{ij}\subset\Gamma$ 是任意一条边,

$$V_n(\partial\Omega_i)=\{\lambda_i\mid\lambda_i|_{\Gamma_{ij}}\in V_n(\Gamma_{ij}),\Gamma_{ij}\subset\partial\Omega_i\},$$

$$V_n(\Gamma)=\{\lambda\mid\lambda|_{\Gamma_{ij}}\in V_n(\Gamma_{ij}),\text{对所有边 }\Gamma_{ij}\subset\Gamma\},$$

$$V_{n\times m}=V_h(\Omega)\times V_n(\Gamma).$$

注4.1:此处不要求网格点的匹配性及 $S_n(\Gamma)$ 中的函数在内交点处的连续性.

以 $0=t_0<t_1<\cdots<t_L=T$ 表区间 $[0,T]$ 的一个一致分划,令 $\tau=t_{j+1}-t_j=T/L$,$t_j=j\tau$.为方便,记

$$v^j=v(x,t_j),\quad j=0,1,\cdots,L;$$

$$v^{j+\frac12}=\frac12(v^{j+1}+v^j),\quad \partial_tv^{j+\frac12}=\frac1\tau(v^{j+1}-v^j),\quad j=0,1,\cdots,L-1.$$

令

$$\langle\lambda,\mu\rangle_{\Gamma_{ij}}=\int_{\Gamma_{ij}}\lambda\mu\mathrm{d}s,\quad\forall\lambda,\mu\in L^2(\Gamma_{ij}),$$

$$\langle \lambda,\mu\rangle_{\partial\Omega_i}=\sum_{\Gamma_{ij}\subset\partial\Omega_i}\langle \lambda,\mu\rangle_{\Gamma_{ij}},\quad \forall\lambda,\mu\in L^2(\Gamma).$$

方程(4-1)的全离散格式是:求$(\{u_h^j\}_{j=0}^L,\{\lambda^j\}_{j=0}^L):[0,T]\to V_{h\times m}$,使

$$\begin{cases}\left(\sum_{i=1}^N\{(\partial_t u_h^{j+\frac12},v)_{\Omega_i}+(\nabla u_h^{j+\frac12},\nabla v)_{\Omega_i}-\langle v,\lambda^{j+\frac12}\rangle_{\partial\Omega_i}\}=\sum_{i=1}^N(f^{j+\frac12},v)_{\Omega_i},\right.\\ \sum_{i=1}^N\langle u_h^{j+1},\mu\rangle_{\partial\Omega_i}=0,\quad\forall(v,\mu)\in V_{h\times n},j=0,1,\cdots,L-1,\\ u_h^0=0,\end{cases}$$

其中μ(或λ)在Γ_{ij}的两侧具相反的符号.λ称为**拉格朗日乘子**.

设$\{\varphi_j^i\}_{j=1}^{N_i}$和$\{\phi_k\}_{k=1}^M$分别是$V_{h_i}(\Omega_i)$和$V_n(\Gamma)$的一组基函数,其中$\phi_k$在$\Gamma_{ij}$的两侧具相反的符号.将内部变量$u_h|_{\Omega_i}$和界面变量$\lambda$在第$k$个时间步的坐标向量分别记作$\boldsymbol{U}_i^k$和$\boldsymbol{x}^k$,函数本身记作$u_{hi}^k$和$\lambda^k$.令

$$\boldsymbol{M}_i=[(\varphi_j^i,\varphi_k^i)_{\Omega_i}]_{N_i\times N_i},\quad \boldsymbol{A}_i=[(\nabla\varphi_j^i,\nabla\varphi_k^i)_{\Omega_i}]_{N_i\times N_i},$$

$$\boldsymbol{B}_i=[\langle\varphi_j^i,\phi_k\rangle_{\partial\Omega_i}]_{N_i\times M},\quad \boldsymbol{F}_i^k=[g_{ij}^k]_{N_i\times 1},$$

其中

$$g_{ij}^k=(\tau f^{(k-1)+\frac12}+u_{hi}^{k-1},\varphi_j^i)_{\Omega_i}-\frac{\tau}{2}(\nabla u_{hi}^{k-1},\nabla\varphi_j^i)_{\Omega_i}+\frac{\tau}{2}\langle\varphi_j^i,\lambda^{k-1}\rangle_{\partial\Omega_i}.$$

则$\boldsymbol{U}_i^k$和$\boldsymbol{x}^k$可按下列方式计算

$$\left(\sum_{i=1}^N\boldsymbol{B}_i^{\mathrm T}(A_i+\frac{2}{\tau}\boldsymbol{M}_i)^{-1}\boldsymbol{B}_i\right)\boldsymbol{x}^k=-\sum_{i=1}^N\boldsymbol{B}_i^{\mathrm T}(\boldsymbol{A}_i+\frac{2}{\tau}\boldsymbol{M}_i)^{-1}\frac{2}{\tau}\boldsymbol{F}_i^k,$$
$$k=1,2,\cdots,L,$$

$$\boldsymbol{U}_i^k=(\boldsymbol{A}_i+\frac{2}{\tau}\boldsymbol{M}_i)^{-1}(\frac{2}{\tau}\boldsymbol{F}_i^k+\boldsymbol{B}_i\boldsymbol{x}^k),\quad i=1,\cdots,N;k=1,\cdots,L.$$

范数$\|\cdot\|_{1,\Omega_i}^2$和$|\cdot|_{-\frac12,\partial\Omega_i}$的定义同第1章.下列结果见文献[30].

定理1　设$u_{tt}\in L^2(0,T;L^2(\Omega_i)),i=1,2,\cdots,N$,且对某个$k\geqslant 2m$有$u\in W^{1,2}(0,T;H^{k+1}(\Omega_i)),i=1,2,\cdots,N$,则

$$\sup_{1\leqslant r\leqslant L}\left\{\|u_h^r-u^r\|_{0,\Omega}+\left(\tau\sum_{j=0}^{r-1}\partial t(u_h^{j+\frac12}-u^{j+\frac12})\|_{0,\Omega}^2\right)^{\frac12}+\right.$$
$$\left(h+\frac{d^{\frac12}}{n}\right)\left(\sum_i\|u_h^r-u^r\|_{1,\Omega_i}^2\right)^{\frac12}+$$
$$\left.\left(h+\frac{d^{\frac12}}{n}\right)\left(\tau\sum_{j=0}^{r-1}\sum_i\left|\lambda^{j+\frac12}-\left(\frac{\partial u}{\partial n}\right)^{j+\frac12}\right|_{-\frac12,\partial\Omega_i}^2\right)^{\frac12}\right\}$$
$$\leqslant c\left(h^m\left(h+\frac{d^{\frac12}}{n}\right)+\tau^2\right)\left(\left(\sum_i\|u\|_{L^2(H^{m+1}(\Omega_i))}^2\right)^{\frac12}+\right.$$
$$\left(\sum_i\|u_t\|_{L^2(H^{k+\frac12}(\partial\Omega_i))}^2\right)^{\frac12}+\left(\sum_i\|u_t\|_{L^2(H^{m+1}(\Omega_i))}^2\right)^{\frac12}+$$

$$\left(\sum_i \| u_t \|^2_{L^2(H^{k+\frac{1}{2}}(\partial\Omega_i))}\right)^{\frac{1}{2}} + \left(\sum_i \| u_{tt} \|^2_{L^2(L^2(L^2(\Omega_i)))}\right)^{\frac{1}{2}}\Bigg)$$

注4.2：定理1中的光滑性假定对某个 $k \geqslant m$ 有“$u \in W^{1,2}(0,T;H^{k+1}(\Omega_i))$, $i = 1,2,\cdots,N$”可以减弱为“$u \in W^{1,2}(0,T;H^{m+1}(\Omega_i))$，且对某个 $k \geqslant 2m$ 有 $u|_{\Gamma} \in W^{1,2}(0,T;H^{k+\frac{1}{2}}(\partial\Omega_i \setminus \partial\Omega))$, $i = 1,2,\cdots,N$”.

4.2　界面矩阵的条件数估计

先给出主要结果.令

$$\overline{S} = \sum_{i=1}^{N} B_i^{\mathrm{T}}(A_i + \frac{2}{\tau}M_i)^{-1}B_i$$

定理2　界面矩阵 $\overline{S}$ 的条件数由下列两式估计

(1) $\mathrm{cond}(\overline{S}) \leqslant C(\tau n^2 d^{-2}), \quad \tau \geqslant d^2$　　(4-2)

(2) $\mathrm{cond}(\overline{S}) \leqslant C(1 + \tau^{\frac{1}{2}} n^2 d^{-1}) d^{-1}, \quad \tau < d^2$　　(4-3)

注4.3：定理2表明界面矩阵 $\overline{S}$ 的条件数与离散时间步长 τ 成正比.但 τ 很小时，$\mathrm{cond}(\overline{S})$ 与 n 无关.对 d 的变化也不太敏感，因此当 N 不太大时，不必构造 $\overline{S}$ 的预条件子(在应用中，不等式“$\tau \geqslant d^2$”或“$\tau < d^2$”不是绝对的关系，d 应理解为 C_0/N，其中 C_0 仅与 Ω 有关)；反之，当 τ 不太小时，必须构造 $\overline{S}$ 的预条件子(见下一节).

为证明定理2，需要几个引理.以下记 $\eta = 2/\tau$.

引理1　设 $\tau < d^2$.若 $u \in H^1(\Omega_i)$，则

$$| u |^2_{\frac{1}{2},\partial\Omega_i} + \eta^{\frac{1}{2}} d^{-1} \| u \|^2_{0,\partial\Omega_i} \leqslant C(| u |^2_{1,\Omega_i} + \eta d^2 \| u \|^2_{0,\Omega_i}). \qquad (4\text{-}4)$$

证明　由标准的相似变换技巧知，只须证明当 $d = 1$ 时(4-4)式成立.

因 $\tau < 1$，故 $\eta^{-1} < 1$.由 ϵ 不等式(1-13)(取 $\epsilon = \eta^{-\frac{1}{4}}$)得

$$\| u \|_{0,\partial\Omega_i} \leqslant C(\eta^{-\frac{1}{4}} | u |_{1,\Omega_i} + \eta^{\frac{1}{4}} \| u \|_{0,\Omega_i}),$$

进一步有

$$\eta^{\frac{1}{2}} \| u \|^2_{0,\partial\Omega_i} \leqslant C(| u |^2_{1,\Omega_i} + \eta \| u \|^2_{0,\Omega_i}).$$

由此及迹定理即知(4-4)式成立($d = 1$).

定义内积 $[\cdot,\cdot]_{\Omega_i}$ 和权范数 $\| \cdot \|^*_{1,\Omega_i}$ 如下：

$$[u,v]_{\Omega_i} = (\nabla u, \nabla v)_{\Omega_i} + \eta(u,v)_{\Omega_i}, \quad \forall u,v \in H^1(\Omega_i),$$

$$\| u \|^*_{1,\Omega_i} = [u,u]^{\frac{1}{2}}_{\Omega_i}, \quad u \in H^1(\Omega_i).$$

引理2　假定 $u \in H^1(\Omega_i)$ 满足

$$[u,\varphi]_{\Omega_i} = 0, \quad \forall \varphi \in H_0^1(\Omega_i),$$

则对所有满足 $v \in H^1(\Omega_i)$ 和 $v|_{\partial\Omega_i} = u|_{\partial\Omega_i}$ 的 v,有

$$\|u\|_{1,\Omega_i}^* \leqslant \|v\|_{1,\Omega_i}^*. \tag{4-5}$$

证明 由 $u - v \in H_0^1(\Omega_i)$ 知 $[u, u-v]_{\Omega_i} = 0$. 故由柯西-施瓦兹不等式得

$$[u,u]_{\Omega_i} = [u,v]_{\Omega_i} \leqslant \|u\|_{1,\Omega_i}^* \cdot \|v\|_{1,\Omega_i}^*.$$

由此即得(4-5)式.

引理 3 若 $\lambda \in V_n(\partial\Omega_i) \cap C(\partial\Omega_i)$,则

$$|\lambda|_{1,\partial\Omega_i} \leqslant Cn^2 d^{-1} |\lambda|_{0,\partial\Omega_i}. \tag{4-6}$$

引理 4 设 $D \subset \mathbf{R}^2$ 是"尺寸"为 d 的多边形,$\varphi \in V_n(\partial D) \cap C(\partial D)$. 假定 $u \in H^1(D)$ 是问题:

$$\begin{cases} -\Delta u + \eta u = 0, & \text{在 } D \text{ 内}, \\ u|_{\partial D} = \varphi \end{cases}$$

的弱解,则

$$\|u\|_{1,D}^{*2} \leqslant C(d^{-1}n^2 + \eta^{\frac{1}{2}} \|\varphi\|_{0,\partial D}^2). \tag{4-7}$$

证明 不失一般性,设 D 是一个四边形. 作一与 D 相似的四边形 $D_0 \subset D$,使 $\mathrm{dist}(\partial D_0, \partial D) = \delta$,其中 $\delta = \min\{\eta^{-\frac{1}{2}}, dn^{-2}\}$. 将 $D \setminus D_0$ 分成若干"尺寸"为 δ 的四边形: $D \setminus D_0 = \bigcup D_i$,使所有分点均落在 ∂D_0 或 ∂D 上. 记 $E_{ij} = \overline{D}_i \cap \overline{D}_j$.

定义 $v \in H^1(D)$ 为下列问题的弱解:

$$\begin{cases} -\Delta v = 0, & \text{在 } D_i \text{ 内}, \\ v|_{\partial D} = \varphi, & v|_{\overline{D}_0} = 0, \end{cases}$$

v 在 E_{ij} 上为线性函数

由引理 2 知 $\|u\|_{1,D}^* \leqslant \|v\|_{1,D}^*$,故只须证

$$\|v\|_{1,D}^{*2} \leqslant C(d^{-1}n^2 + \eta^{\frac{1}{2}}) \|\varphi\|_{0,\partial D}^2). \tag{4-8}$$

由庞加莱不等式,知

$$\delta^{-2} \|v\|_{0,D_i}^2 \leqslant C(|v|_{1,D_i}^2 + \delta^{-1} \|v\|_{0,\partial D_i}^2),$$

故

$$\|v\|_{0,D_i}^2 \leqslant C(\delta^2 |v|_{1,D_i}^2 + \delta \|v\|_{0,\partial D_i}^2),$$

因此

$$\|v\|_{1,D_i}^2 \leqslant C(|v|_{1,D_i}^2 + \eta^{\frac{1}{2}} \|v\|_{0,\partial D_i}^2). \tag{4-9}$$

另一方面,由 v 的定义有(见(1-17)式)

$$|v|_{1,D_i} \leqslant C|v|_{\frac{1}{2},\partial D_i},$$

由此及(4-9)式得

$$\|v\|_{1,D_i}^{*2} \leqslant C(|v|_{\frac{1}{2},\partial\Omega_i}^2 + \eta^{\frac{1}{2}} \|v\|_{0,\partial D_i}^2). \tag{4-10}$$

利用内插公式(1-12),有

$$|v|_{\frac{1}{2},\partial D_i}^2 \leqslant C(\delta|v|_{1,\partial D_i}^2+\delta^{-1}\|v\|_{0,\partial D_i}^2)^{\frac{1}{2}}\cdot\delta^{-\frac{1}{2}}\|v\|_{0,\partial D_i}$$

$$\leqslant C(|v|_{1,\partial D_i}\cdot\|v\|_{0,\partial D_i}+\delta^{-1}\|v\|_{0,\partial D_i}^2). \tag{4-11}$$

以 x_{ij} 表 ∂D 与 E_{ij} 的交点. 由 v 的定义有

$$|v|_{0,E_{ij}}^2=\frac{1}{2}\delta v^2(x_{ij}),\quad |v|_{1,E_{ij}}^2=\delta^{-1}v^2(x_{ij}),$$

故

$$\|v\|_{0,\partial D_i}^2\leqslant C(\|v\|_{0,\partial D\cap\partial D_i}^2+\delta\sum_j v^2(x_{ij})), \tag{4-12}$$

$$|v|_{1,\partial D_i}^2\leqslant C(|v|_{1,\partial D\cap\partial D_i}^2+\delta^{-1}\sum_j v^2(x_{ij})). \tag{4-13}$$

定义

$$S_\delta(\partial D)=\{\psi\mid\psi\in C(\partial D),\text{对每个 }i,\psi|_{\partial D\cap\partial D_i}\text{ 是线性的}\},$$

并以 $\pi_\delta: C(\partial D)\to S_\delta(\partial D)$ 表相应的插值算子,使

$$\pi_\delta\varphi(x_{ij})=\varphi(x_{ij}),\quad \forall\varphi\in C(\partial D)\text{ 及任何 } i,j,$$

则有

$$\|(\pi_\delta-I)v\|_{0,\partial D\cap\partial D_i}\leqslant C\delta|v|_{1,\partial D\cap\partial D_i}.$$

利用有限元函数的等价离散范数,进一步得到

$$\delta\sum_{i,j}v^2(x_{ij})\leqslant C\|\pi_\delta v\|_{0,\partial D}^2=C(\|v\|_{0,\partial D}^2+\|(\pi_\delta-I)v\|_{0,\partial D}^2)$$

$$\leqslant C(\|v\|_{0,\partial D}^2+\delta^2|v|_{1,\partial D}^2).$$

由此及(4-12)和(4-13)式推得

$$\sum_i\|v\|_{0,\partial D_i}^2\leqslant C(\|v\|_{0,\partial D}^2+\delta^2|v|_{1,\partial D}^2), \tag{4-14}$$

$$\sum_i|v|_{1,\partial D_i}^2\leqslant C(|v|_{1,\partial D}^2+\delta^{-2}|v|_{0,\partial D}^2).$$

因此,由(4-11)式和柯西不等式得

$$\sum_i|v|_{\frac{1}{2},\partial D_i}^2\leqslant C(\delta|v|_{1,\partial D}^2+|v|_{1,\partial D}\|v\|_{0,\partial D}+\delta^{-1}\|v\|_{0,\partial D}^2).$$

由此及(4-6)、(4-10)、(4-14)式即证得(4-8)式.

引理5 对任何 $\lambda\in V_n(\partial\Omega_i)$,存在 $v_0\in S_n(\partial\Omega_i)\cap C(\partial\Omega_i)$,使

$$|\langle\lambda,v_0\rangle_{\partial\Omega_i}|\geqslant C\|\lambda\|_{0,\partial\Omega_i}\cdot\|v\|_{0,\partial\Omega_i}. \tag{4-15}$$

现给出定理2的证明.定义算子

$$\mathrm{R}_i: V_n(\partial\Omega_i)\to V_{hi}(\Omega_i)\text{ 和 }\mathrm{S}_i: V_n(\Gamma)\to V_n(\Gamma)\text{ 如下:}$$

$$[\mathrm{R}\lambda_i,v]_{\Omega_i}=\langle\lambda_i,v\rangle_{\partial\Omega_i},\quad \lambda_i\in V_n(\partial\Omega_i),\forall v\in V_{hi}(\Omega_i),$$

$$\langle\mathrm{S}\lambda,\mu\rangle_{\partial\Omega_i}=[\mathrm{R}\lambda,\mathrm{R}_i u]_{\Omega_i},\quad \lambda\in V_n(\Gamma),\forall\mu\in V_n(\Gamma).$$

令 $\mathrm{S}=\sum_{i=1}^N\mathrm{S}_i$,易知 S 就是 $\overline{S}$ 所对应的算子(即 $\overline{S}$ 是 S 的刚度矩阵).

对 $v\in H^{\frac{1}{2}}(\partial\Omega_i)$,以 $w_v\in H^1(\Omega_i)$ 表下列问题的弱解:

$$\begin{cases} -\Delta w_v + \eta w_v = 0, & \text{在 } \Omega_i \text{ 内}, \\ w_v|_{\partial\Omega_i} = v. \end{cases}$$

定义

$$|\lambda|^*_{-\frac{1}{2},\partial\Omega_i} = \sup_{v\in H^{\frac{1}{2}}(\partial\Omega_i)} \frac{|\langle\lambda, v\rangle_{\partial\Omega_i}|}{\|v\|^*_{\frac{1}{2},\partial\Omega_i}},$$

其中 $\|v\|^*_{\frac{1}{2},\partial\Omega_i} = \|w_v\|^*_{1,\Omega_i}$.

由 R_i 的定义和引理 2 知

$$\|\mathrm{R}_i\lambda\|^{*2}_{1,\Omega_i} = [\mathrm{R}_i\lambda, \mathrm{R}_i\lambda]_{\Omega_i} = \langle \mathrm{R}_i\lambda, \lambda\rangle_{\partial\Omega_i}$$
$$\leqslant \|\mathrm{R}_i\lambda\|^*_{\frac{1}{2},\partial\Omega_i} |\lambda|^*_{-\frac{1}{2},\partial\Omega_i} = \|\mathrm{R}_i\lambda\|^*_{1,\Omega_i} |\lambda|^*_{-\frac{1}{2},\partial\Omega_i},$$

故

$$\|\mathrm{R}_i\lambda\|^*_{1,\Omega_i} \leqslant |\lambda|^*_{-\frac{1}{2},\partial\Omega_i}. \tag{4-16}$$

当 $\eta > d^{-2}$ 时，由引理 1 知，对任何 $v \in H^{\frac{1}{2}}(\partial\Omega_i)$ 有

$$\|w_v\|^{*2}_{1,\Omega_i} = d^2(d^{-2}|w_v|^2_{1,\Omega_i} + \eta d^{-2}\|w_v\|^2_{0,\Omega_i})$$
$$\geqslant Cd^2(|w_v|^2_{1,\Omega_i} + \eta d^{-2}\|w_v\|^2_{0,\Omega_i}) \quad (\text{因 } d \text{ 有上界})$$
$$\geqslant Cd^2(|v|^2_{\frac{1}{2},\partial\Omega_i} + \eta^{\frac{1}{2}} d^{-1}\|v\|^2_{0,\partial\Omega_i})$$
$$\geqslant Cd\eta^{\frac{1}{2}}\|v\|^2_{0,\partial\Omega_i}$$

因此

$$\|\lambda\|^{*2}_{-\frac{1}{2},\partial\Omega_i} \leqslant C\eta^{-\frac{1}{2}} d^{-1}\|\lambda\|^2_{0,\partial\Omega_i}.$$

由此及(4-16)式得

$$\langle \mathrm{S}\lambda, \lambda\rangle = \sum_{i=1}^{N} \|\mathrm{R}_i\lambda\|^{*2}_{1,\Omega_i} \leqslant C\eta^{-\frac{1}{2}} d^{-1}\|\lambda\|^2_{0,\Gamma} (\eta > d^{-2}). \tag{4-17}$$

当 $\eta \leqslant d^{-2}$ 时，由迹不等式知，对任何 $v \in H^{\frac{1}{2}}(\partial\Omega_i)$ 有

$$\|w_v\|^{*2}_{1,\Omega_i} = |w_v|^2_{1,\Omega_i} + \eta\|w_v\|^2_{0,\Omega_i}$$
$$= \eta d^2(\eta^2 d^{-2}|w_v|^2_{1,\Omega_i} + d^{-2}\|w_v\|^2_{0,\Omega_i})$$
$$\geqslant \eta d^2(|w_v|^2_{1,\Omega_i} + d^{-2}\|w_v\|^2_{0,\Omega_i}) \quad (\text{因 } \eta^{-1}d^{-2} \geqslant 1)$$
$$\geqslant \eta d^2\|v\|^2_{\frac{1}{2},\partial\Omega_i} \geqslant \eta d^*\|v\|^2_{0,\partial\Omega_i}$$

因此

$$|\lambda|^{*2}_{-\frac{1}{2},\partial\Omega_i} \leqslant \eta^{-1}d^{-1}\|\lambda\|^2_{0,\partial\Omega_i}.$$

由此及(4-16)式得

$$\langle \mathrm{S}\lambda, \lambda\rangle \leqslant \eta^{-1}d^{-1}\|\lambda\|^2_{0,\Gamma} \quad (\eta \leqslant d^{-2}). \tag{4-18}$$

另一方面，以 $P_h: H^1(\Omega_i) \to V_h(\Omega_i)$ 表通常的 L^2 投影算子，则对引理5中的 v_0 有

$$
\begin{aligned}
|\langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i}| &= |[R\lambda, P_h w_{v_0}]_{\Omega_i}| \\
&\leqslant \| R\lambda \|^*_{1,\Omega_i^0} \cdot \| P_h w_{v_0} \|^*_{1,\Omega_i} \\
&\leqslant \| w_{v_0} \|^*_{1,\Omega_i} \cdot \| R\lambda \|^*_{1,\Omega_i},
\end{aligned}
$$

由引理4，进一步得

$$
|\langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i}| \leqslant C(d^{-1}n^2 + \eta^{\frac{1}{2}})^{\frac{1}{2}} \| v_0 \|_{0,\partial\Omega_i} \cdot \| R\lambda \|^*_{1,\Omega_i}. \tag{4-19}
$$

若能证明

$$
\| \lambda \|_{0,\partial\Omega_i} \| v_0 \|_{0,\partial\Omega_i} \leqslant C\langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i}, \tag{4-20}
$$

则由此及(4-19)式，立得

$$
\| \lambda \|_{0,\partial\Omega_i} \leqslant C(d^{-1}n^2 + \eta^{\frac{1}{2}})^{\frac{1}{2}} \| R\lambda \|^*_{1,\Omega_i}
$$

即

$$
\frac{C \| \lambda \|^2_{0,F}}{d^{-1}n^2 + \eta^{\frac{1}{2}}} \leqslant \langle S\lambda, \lambda \rangle. \tag{4-21}
$$

特别，当 $\eta \leqslant d^{-2}$ 时，有

$$
Cn^{-2}d \| \lambda \|^2_{0,\Gamma} \leqslant \langle S\lambda, \lambda \rangle.
$$

由此及(4-17)、(4-18)、(4-21)式可推得定理2.

由引理5，要证(4-20)式，只须证

$$
\langle \lambda, v_0 \rangle_{\partial\Omega_i} \leqslant C\langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i}. \tag{4-22}
$$

事实上，由 ϵ 不等式(取 $\epsilon = h^{1/2}$)得

$$
\begin{aligned}
\| w_{v_0} - P_h w_{v_0} \|_{0,\partial\Omega_i} &\leqslant C(h^{1/2} | w_{v_0} - P_h w_{v_0} |_{1,\Omega_i} + \\
&\qquad h^{-\frac{1}{2}} \| w_{v_0} - P_h w_{v_0} \|_{0,\Omega_i}) \\
&\leqslant Ch^{\frac{1}{2}}(| w_{v_0} |_{1,\Omega_i} + h^{-1} \| w_{v_0} - P_h w_{v_0} \|_{0,\Omega_i}) \\
&\leqslant Ch^{\frac{1}{2}} | w_{v_0} |_{1,\Omega_i}.
\end{aligned}
$$

利用引理5，进一步有

$$
\| v_0 - P_h w_{v_0} \|_{0,\partial\Omega_i} \leqslant Ch^{\frac{1}{2}}(n^2 + \eta^{\frac{1}{2}})^{\frac{1}{2}} \| v_0 \|_{0,\partial\Omega_i}.
$$

故由引理4，得

$$
\begin{aligned}
\langle \lambda, v_0 \rangle_{\partial\Omega_i} &\leqslant \langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i} + \langle \lambda, P_h w_{v_0} - v_0 \rangle_{\partial\Omega_i} \\
&\leqslant \langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i} + \| \lambda \|_{0,\partial\Omega_i} \| P_h w_{v_0} - v_0 \|_{0,\partial\Omega_i} \\
&\leqslant \langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i} + Ch^{\frac{1}{2}}(n^2 + \eta^{\frac{1}{2}})^{\frac{1}{2}} \| \lambda \|_{0,\partial\Omega_i} \| v_0 \|_{0,\partial\Omega_i} \\
&\leqslant \langle \lambda, P_h w_{v_0} \rangle_{\partial\Omega_i} + Ch^{\frac{1}{2}}(n^2 + \eta^{\frac{1}{2}})^{\frac{1}{2}} \langle \lambda, v_0 \rangle_{0,\partial\Omega_i}.
\end{aligned}
$$

由此及假定 H_4(自然可设 $h\eta^{\frac{1}{2}} \to 0$) 即证得(4-22)式.

4.3 界面预条件子的构造

先给出主要结果.以$\{\mu_r^{(ij)}\}_{r=1}^{n+1}$表$V_n(\Gamma_{ij})(i<j)$的一组基函数.记

$$a_{kr}^{(ij)}=\langle\varphi_k^i,\mu_r^{(ij)}\rangle_{\Gamma_{ij}},\quad r=1,\cdots,n+1;k=1,\cdots,N_i,$$

$$a_{kr}^{(ji)}=\langle\varphi_k^j,\mu_r^{(ij)}\rangle_{\Gamma_{ij}},\quad r=1,\cdots,n+1;k=1,\cdots,N_j,$$

所构成的矩阵分别记作$\boldsymbol{B}_{ij}$和$\boldsymbol{B}_{ji}$.令

$$\overline{\boldsymbol{S}}_{ij}=\boldsymbol{B}_{ij}^{\mathrm{T}}(\boldsymbol{A}_i+\frac{2}{\tau}\boldsymbol{M}_i)^{-1}\boldsymbol{B}_{ij}+\boldsymbol{B}_{ji}^{\mathrm{T}}(\boldsymbol{A}_i+\frac{2}{\tau}\boldsymbol{M}_j)^{-1}\boldsymbol{B}_{ji},$$

将边的集合$\{\Gamma_{ij}\mid\Gamma_{ij}\subset\partial\Omega_i,\partial\Omega_i$满足$\partial\Omega_i\cap\partial\Omega=\varnothing\}$写成$\{\Gamma_{or}\}_{r=1}^{N_0}$,以$\mathrm{D}_{or}:V_n(\Gamma_{or})\to V_n(\Gamma)$表通常的零扩张算子.记

$$b_{kr}^{(oi)}=\langle\varphi_k^i,\mathrm{D}_{or}1\rangle_{\partial\Omega_i},\quad k=1,\cdots,N_i;r=1,\cdots,N_0,$$

以它们为元素构成的矩阵记作$\boldsymbol{B}_{oi}$.令

$$\overline{\boldsymbol{S}}_0=\sum_{i=1}^{N}\boldsymbol{B}_{oi}^{\mathrm{T}}(\boldsymbol{A}_i+\frac{2}{\tau}\boldsymbol{M}_i)^{-1}\boldsymbol{B}_{oi}.$$

定义$\overline{S}$的预条件子为

$$\overline{\boldsymbol{M}}_1=\sum_{i<j}\overline{\boldsymbol{D}}_{ij}\overline{\boldsymbol{S}}_{ij}^{-1}\overline{\boldsymbol{D}}_{ij}^{\mathrm{T}}\quad(\text{若 } d\geqslant\eta^{-\frac{1}{2}})$$

或

$$\overline{\boldsymbol{M}}_2=\boldsymbol{I}_0\overline{\boldsymbol{S}}_0^{-1}\boldsymbol{I}_0^{\mathrm{T}}+\overline{\boldsymbol{M}}_1\quad(\text{若 } d<\eta^{-\frac{1}{2}}).$$

此处$\overline{\boldsymbol{D}}_{ij}$表零扩张算子$\mathrm{D}_{ij}$所对应的0、1符号矩阵,$\boldsymbol{I}_0$是按如下方式定义的0、1符号矩阵

$$(\mathrm{D}_{01}1,\cdots,\mathrm{D}_{0N_0}1)=(\psi_1,\cdots,\psi_M)\boldsymbol{I}_0.$$

定理 3 1° 若$\tau<d^2$,则

$$\mathrm{cond}(\overline{\boldsymbol{M}}_1,\overline{\boldsymbol{S}})\leqslant C(1+\log^2 n);$$

2° 若$\tau\geqslant d^2$,则

$$\mathrm{cond}(\overline{\boldsymbol{M}}_2,\overline{\boldsymbol{S}})\leqslant C(1+\log^2 n).$$

注 4.4: 定理3表明,当时间步长τ相对于d较小时,粗可解子是不必要的,即只要用块对解矩阵$\overline{\boldsymbol{M}}_1$作为预条件子,就可大大改善界面矩阵的条件数.此外,我们构造的预条件子具结构简单(不需考虑内交点)、计算量小的优点.

为证明定理3,需要若干引理.

引理 6 对$\lambda\in S_n(\partial\Omega_i)$,有

$$|\lambda|_{0,\partial\Omega_i}\leqslant Cnd^{-\frac{1}{2}}|\lambda|_{-\frac{1}{2},\partial\Omega_i}.\tag{4-23}$$

引理 7 假定$\lambda\in V_n(\Gamma)$,则

$$|\lambda|_{-\frac{1}{2},\partial\Omega_i}^2\leqslant C\langle\mathrm{S}_i\lambda,\lambda\rangle\quad(\text{若 }\eta\leqslant d^{-2})\tag{4-24}$$

和

$$|\lambda|_{-\frac{1}{2},\partial\Omega_i}^{*2}\leqslant C\langle\mathrm{S}_i\lambda,\lambda\rangle.\tag{4-25}$$

其中 $\eta = \frac{2}{\tau}$(下同). 注意当 $\eta > d^{-2}$ 时(4-24)式不成立.

证明 只证(4-25)式,(4-24)式的证明类似. 以 $u_\lambda \in H^1(\Omega_i)$ 表下列问题的弱解:

$$\begin{cases} -\Delta u_\lambda + \eta u_\lambda = 0, & 在\ \Omega_i\ 内, \\ \dfrac{\partial u_\lambda}{\partial n}\Big|_{\partial\Omega_i} = \lambda. \end{cases}$$

由 $|\lambda|^*_{-\frac{1}{2},\partial\Omega_i}$ 的定义,有

$$|\lambda|^*_{-\frac{1}{2},\partial\Omega_i} = \sup_{v\in H^{\frac{1}{2}}(\partial\Omega_i)} \frac{|\langle\lambda, v\rangle_{\partial\Omega_i}|}{\|v\|^*_{\frac{1}{2},\partial\Omega_i}} = \sup_{v\in H^{\frac{1}{2}}(\partial\Omega_i)} \frac{|(u_\lambda, w_v)_{\Omega_i}|}{\|w_v\|^*_{1,\Omega_i}}$$

$$\leqslant \sup_{v\in H^{\frac{1}{2}}(\partial\Omega_i)} \frac{\|u_\lambda\|^*_{1,\Omega_i}\cdot\|w_v\|^*_{1,\Omega_i}}{\|w_v\|^*_{1,\Omega_i}} = \|u_\lambda\|^*_{1,\Omega_i},$$

故

$$\|u_\lambda\|^*_{1,\Omega_i}\cdot|\lambda|^*_{-\frac{1}{2},\partial\Omega_i} \leqslant \langle u_\lambda, \lambda\rangle_{\partial\Omega_i}. \tag{4-26}$$

以 $u_{h\lambda} \in S_{h_i}(\Omega_i)$ 表 u_λ 的 L^2 投影,则有

$$|u_{h\lambda}|_{1,\Omega_i} \leqslant |u_\lambda|_{1,\Omega_i}, \quad \|u_{h\lambda} - u_\lambda\|_{0,\Omega_i} \leqslant Ch\,|u_\lambda|_{1,\Omega_i}. \tag{4-27}$$

故由 ε 不等式(1-13)得

$$\|u_{h\lambda} - u_\lambda\|_{0,\partial\Omega_i} \leqslant C(h^{\frac{1}{2}}|u_{h\lambda} - u_\lambda|_{1,\Omega_i} + h^{-\frac{1}{2}}\|u_{h\lambda} - u_\lambda\|_{0,\Omega_i})$$

$$\leqslant Ch^{\frac{1}{2}}|u_\lambda|_{1,\Omega_i} \leqslant Ch^{\frac{1}{2}}\|u_\lambda\|^*_{1,\Omega_i}. \tag{4-28}$$

又由引理4、引理5知

$$\|\lambda\|_{0,\partial\Omega_i}\cdot\|v_0\|_{0,\partial\Omega_i} \leqslant C\langle\lambda, v_0\rangle_{\partial\Omega_i} = C[u_\lambda, w_{v_0}]_i$$

$$\leqslant C\|u_\lambda\|^*_{1,\Omega_i}\cdot\|w_{v_0}\|_{1,\Omega_i}$$

$$\leqslant C(n^2d^{-1} + \eta^{\frac{1}{2}})^{\frac{1}{2}}\|v_0\|_{0,\partial\Omega_i}\cdot\|u_\lambda\|^*_{1,\Omega_i},$$

即
$$\|\lambda\|_{0,\partial\Omega_i} \leqslant C(n^2d^{-1} + \eta^{\frac{1}{2}})^{\frac{1}{2}}\|u_\lambda\|^*_{1,\Omega_i}.$$

由此及(4-28)式得

$$\langle u_\lambda, \lambda\rangle_{\partial\Omega_i} = \langle u_{h\lambda}, \lambda\rangle_{\partial\Omega_i} + \langle u_{h\lambda} - u_\lambda, \lambda\rangle_{\partial\Omega_i}$$

$$\leqslant \langle u_{h\lambda}, \lambda\rangle_{\partial\Omega_i} + \|u_{h\lambda} - u_\lambda\|_{0,\partial\Omega_i}\|\lambda\|_{0,\partial\Omega_i}$$

$$\leqslant \langle u_{h\lambda}, \lambda\rangle_{\partial\Omega_i} + Ch^{\frac{1}{2}}(n^2 + \eta^{\frac{1}{2}})^{\frac{1}{2}}\langle u_\lambda, \lambda\rangle_{\partial\Omega_i},$$

故由 H_4,有(设 $h\eta^{\frac{1}{2}} \longrightarrow 0$)

$$\langle u_\lambda, \lambda\rangle_{\partial\Omega_i} \leqslant C\langle u_{h\lambda}, \lambda\rangle_{\partial\Omega_i}. \tag{4-29}$$

另一方面,由(4-27)式又有

$$\|u_{h\lambda}\|^*_{1,\Omega_i} \leqslant C\|u_\lambda\|^*_{1,\Omega_i}$$

由此及(4-26)、(4-29)式推得

$$\| u_{h\lambda} \|_{1,\Omega_i}^* \cdot | \lambda |_{-\frac{1}{2},\partial\Omega_i}^* \leqslant C\langle u_{h\lambda}, \lambda\rangle_{\partial\Omega_i} = C[u_{h\lambda}, R_i\lambda]_{\Omega_i}$$
$$\leqslant C \| u_{h\lambda} \|_{1,\Omega_i}^* \cdot \| R_i\lambda \|_{1,\Omega_i}^*,$$

即

$$| \lambda |_{-\frac{1}{2},\partial\Omega_i}^* \leqslant C \| R_i\lambda \|_{1,\Omega_i}^*.$$

因此(4-25)式成立.

引理 8　对 $v \in H^{\frac{1}{2}}(F_{ij})$,有

$$\| v - \gamma_{F_{ij}}(v) \|_{\frac{1}{2},F_{ij}} \leqslant C | v |_{\frac{1}{2},\Gamma_{ij}} \tag{4-30}$$

证明　此处可用初等方法的证明.不失一般性,设 $\Gamma_{ij} = \{(x,0) \mid 0 \leqslant x \leqslant d\}$.记 $\tilde{\Omega} = [0,d]^2$,并以 $\bar{v} \in H^{\frac{1}{2}}(\partial\tilde{\Omega})$ 表由下列方式定义的函数

$$\bar{v}(x,y) = \begin{cases} v(x,0), & 0 \leqslant x \leqslant d, y = 0, \\ v(d-x,0), & 0 \leqslant x \leqslant d, y = d, \\ v(y,0), & 0 \leqslant y \leqslant d, x = 0, \\ v(d-y,0), & 0 \leqslant y \leqslant d, x = d, \end{cases}$$

即空间曲线 $Z = \bar{v}(x,y)((x,y) \in \partial\tilde{\Omega})$ 关于平面 $x = y$ 和平面 $x + y = d$ 对称.先验证

$$| \bar{v} |_{\frac{1}{2},\partial\tilde{\Omega}} \leqslant C | v |_{\frac{1}{2},\Gamma_{ij}}. \tag{4-31}$$

为方便起见,记 $\Gamma_{ij} = E_1$,$\partial\tilde{\Omega}$ 的其它边按逆时针方向依次记作 E_2,E_3 和 E_4,则

$$| \bar{v} |_{\frac{1}{2},\partial\tilde{\Omega}}^2 = \sum_{k=1}^4 \sum_{r=1}^4 \int_{E_k}\int_{E_r} \frac{| \bar{v}(A) - \bar{v}(B) |^2}{| A - B |^2} \mathrm{d}s(A)\mathrm{d}s(B)$$

其中 A、B 分别表 E_k 和 E_r 上的点.要证(4-31)式,只须证

$$\int_{E_k}\int_{E_r} \frac{| \bar{v}(A) - \bar{v}(B) |^2}{| A - B |^2} \mathrm{d}s(A)\mathrm{d}s(B) \leqslant | v |_{\frac{1}{2},E_1}^2,$$

其中 k、r 不全为 1.不妨取 $k = 1$,$r = 2$,则

$$\int_{E_1}\int_{E_2} \frac{| \bar{v}(A) - \bar{v}(B) |^2}{| A - B |^2} \mathrm{d}s(A)\mathrm{d}s(B) = \int_0^d\int_0^d \frac{| \bar{v}(x,0) - \bar{v}(0,y) |^2}{(d-x)^2 + y^2} \mathrm{d}x\mathrm{d}y$$
$$= \int_0^d\int_0^d \frac{| v(x,0) - v(d-y,0) |^2}{(d-x)^2 + y^2} \mathrm{d}x\mathrm{d}y \leqslant \int_0^d\int_0^d \frac{| v(x,0) - v(d-y,0) |^2}{| d - x - y |^2} \mathrm{d}x\mathrm{d}y$$
$$= \int_0^d\int_0^d \frac{| v(x,0) - v(y,0) |^2}{| x - y |^2} \mathrm{d}x\mathrm{d}y = | v |_{\frac{1}{2},E_1}^2.$$

以 $\tilde{v} \in H^1(\tilde{\Omega})$ 表 $\bar{v}$ 的调和扩张,即 $\tilde{v}$ 是下列问题的弱解:

$$\begin{cases} -\Delta\tilde{v} = 0, & \text{在 } \tilde{\Omega} \text{ 内}, \\ \tilde{v}|_{\partial\tilde{\Omega}} = \bar{v}. \end{cases}$$

则有

$$|\tilde{v}|_{1,\Omega} \leqslant C|\bar{v}|_{\frac{1}{2},\partial\tilde{\Omega}},$$

故由迹定理和弗里德利希不等式(1-11)得

$$\|v-\gamma_{\Gamma_{ij}}(v)\|_{\frac{1}{2},\Gamma_{ij}} \leqslant \|\bar{v}-\gamma_{\Gamma_{ij}}(v)\|_{\frac{1}{2},\partial\tilde{\Omega}}$$

$$\leqslant C(|\tilde{v}-\gamma_{\Gamma_{ij}}(v)|_{1,\tilde{\Omega}}^2 + d^{-2}\|\tilde{v}-\gamma_{\Gamma_{ij}}(v)\|_{0,\tilde{\Omega}}^2)^{\frac{1}{2}}$$

$$\leqslant C|\tilde{v}|_{1,\tilde{\Omega}} \leqslant C|\bar{v}|_{\frac{1}{2},\partial\tilde{\Omega}}.$$

由此及(4-31)式即证得(4-30)式.

引理9 设 $\lambda_i \in V_n(\partial\Omega_i)$,记 $\lambda_{ij} = \lambda_i|_{\Gamma_{ij}}$,则有

$$|\lambda_{ij}|_{-\frac{1}{2},\Gamma_{ij}} \leqslant C(1+\log n)|\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \tag{4-32}$$

$$|D_{ij}\lambda_{ij}|_{-\frac{1}{2},\Gamma_{ij}}^* \leqslant C(1+\log n)|\lambda_i|_{-\frac{1}{2},\partial\Omega_i}^* \quad (\text{若 } d \geqslant \eta^{-\frac{1}{2}}). \tag{4-33}$$

证明 将 $\partial\Omega_i$ 作直径为 $\bar{h} = dn^{-2}$ 的拟一致剖分,使其顶点为分点.以 $V_{\bar{h}}(\partial\Omega_i)$ 和 $S_{\bar{h}}(\Gamma_{ij})$ 分别表定义在 $\partial\Omega_i$ 上和 Γ_{ij} 上的连续分片线性函数空间.对 $v \in H^{\frac{1}{2}}(\Gamma_{ij})$,存在 $v_h \in V_{\bar{h}}(\Gamma_{ij})$,使

$$\|v_h - v\|_{0,\Gamma_{ij}} \leqslant C\bar{h}^{\frac{1}{2}}\|v\|_{\frac{1}{2},\Gamma_{ij}}, \quad \|v_h\|_{\frac{1}{2},\Gamma_{ij}} \leqslant C\|v\|_{\frac{1}{2},\Gamma_{ij}}, \tag{4-34}$$

故由引理6有

$$\begin{aligned} |\langle\lambda_{ij}, v-v_h\rangle_{\Gamma_{ij}}| &\leqslant \|\lambda_{ij}\|_{0,\Gamma_{ij}} \cdot \|v-v_h\|_{0,\Gamma_{ij}} \\ &\leqslant C\bar{h}^{\frac{1}{2}}\|v\|_{\frac{1}{2},\Gamma_{ij}} \cdot \|\lambda_i\|_{0,\partial\Omega_i} \\ &\leqslant C\|v\|_{\frac{1}{2},\Gamma_{ij}} \cdot \|\lambda_i\|_{-\frac{1}{2},\partial\Omega_i}. \end{aligned} \tag{4-35}$$

设 s_1 和 s_4 是 Γ_{ij} 的两个端点,s_2 和 s_3 分别是与 s_1 和 s_4 相邻的内结点.记 $e = [s_1, s_2] \cup [s_3, s_4]$.定义 $v_{1h} \in V_{\bar{h}}(\Gamma_{ij})$ 和 $v_{2h} \in V_{\bar{h}}(\partial\Omega_i)$ 如下:

$$v_{1h}(s) = \begin{cases} v_h(s), & s = s_1 \text{ 或 } s = s_4, \\ \text{线性函数}, & s \in e, \\ 0, & s \in [s_2, s_3], \end{cases} \qquad v_{2h} = \begin{cases} v_h - v_{1h}, & \text{在 } \Gamma_{ij} \text{ 上}, \\ 0, & \text{在 } \partial\Omega_i \setminus \Gamma_{ij}. \end{cases}$$

由于(见(1-15)式)

$$\|v_h\|_{0,\infty,\Gamma_{ij}} \leqslant C(1+\log^{\frac{1}{2}}(d/\bar{h}))\|v_h\|_{\frac{1}{2},\Gamma_{ij}},$$

故由(4-23)式,得

$$\begin{aligned} |\langle\lambda_{ij}, v_{1h}\rangle_{\Gamma_{ij}}| &\leqslant \|\lambda_{ij}\|_{0,\Gamma_{ij}} \cdot \|v_{1h}\|_{0,\Gamma_{ij}} \\ &\leqslant \|\lambda_i\|_{0,\partial\Omega_i} \cdot \|v_{1h}\|_{0,e} \\ &\leqslant Cnd^{-\frac{1}{2}}|\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \cdot \bar{h}^{\frac{1}{2}}\|v_h\|_{0,\infty,\Gamma_{ij}} \\ &\leqslant C(1+\log^{\frac{1}{2}} n)|\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \cdot \|v_h\|_{\frac{1}{2},\Gamma_{ij}}. \end{aligned} \tag{4-36}$$

由(1-16)式,知

$$\| v_{2h} \|_{\frac{1}{2},\partial\Omega_i} \leqslant C(1+\log(d/\overline{h})) \| v_h \|_{\frac{1}{2},\Gamma_{ij}},$$

因此

$$|\langle \lambda_i, v_{2h}\rangle_{\partial\Omega_i}| \leqslant |\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \cdot \| v_{2h} \|_{\frac{1}{2},\partial\Omega_i} \tag{4-37}$$
$$\leqslant C(1+\log n)|\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \cdot \| v_h \|_{\frac{1}{2},\Gamma_{ij}}.$$

最后,由(4-34),(4-35),(4-36)和(4-37)式,得到

$$\begin{aligned}|\langle \lambda_{ij}, v\rangle_{\Gamma_{ij}}| &\leqslant |\langle \lambda_{ij}, v_h\rangle_{\Gamma_{ij}}| + |\langle \lambda_{ij}, v - v_h\rangle_{\Gamma_{ij}}| \\ &\leqslant |\langle \lambda_{ij}, v_{1h}\rangle_{\Gamma_{ij}}| + |\langle \lambda_i, v_{2h}\rangle_{\partial\Omega_i}| + \\ &\quad C\| v \|_{\frac{1}{2},\Gamma_{ij}} \cdot \| \lambda_i \|_{-\frac{1}{2},\partial\Omega_i} \\ &\leqslant C(1+\log n)|\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \cdot \| v \|_{\frac{1}{2},\Gamma_{ij}}\end{aligned}$$

此即(4-32)式(注意 v 的任意性).

现考虑(4-33)式,从(4-32)式的证明可看出

$$|\langle \lambda_{ij}, v\rangle_{\Gamma_{ij}}| \leqslant C(1+\log n)|\lambda_i|_{-\frac{1}{2},\partial\Omega_i} \cdot \| v \|_{\frac{1}{2},\partial\Omega_i}, \quad \forall v \in H^{\frac{1}{2}}(\partial\Omega_i).$$

故只须证:对任意 $v \in H^{\frac{1}{2}}(\partial\Omega_i)$,有

$$\| v \|_{\frac{1}{2},\partial\Omega_i} \leqslant C\| v \|^{*}_{\frac{1}{2},\partial\Omega_i}, \quad 当\ d \geqslant \eta^{-\frac{1}{2}}\ 时.$$

事实上,设 $\varphi_v \in H^1(\Omega_i)$ 是下列问题的弱解:

$$\begin{cases} -\Delta\varphi_2 + d^{-2}\varphi_v = 0, & 在\ \Omega_i\ 内, \\ \varphi_v|_{\partial\Omega_i} = v, \end{cases}$$

则由迹定理和熟知的"最小性"结果(将引理2中的 η 换成 d^{-2})得

$$\begin{aligned}\| v \|_{\frac{1}{2},\partial\Omega_i} &\leqslant C(|\varphi_v|^2_{1,\Omega_i} + d^{-2}\| \varphi_v \|^2_{0,\Omega_i})^{\frac{1}{2}} \\ &\leqslant C(|w_v|^2_{1,\Omega_i} + d^{-2}\| w_v \|^2_{0,\Omega_i})^{\frac{1}{2}} \\ &\leqslant C(|w_v|^{*}_{1,\Omega_i} = C\| v \|^{*}_{\frac{1}{2},\partial\Omega_i} \quad (因\ d^{-2} \leqslant \eta).\end{aligned}$$

定义算子 $\mathrm{R}_{ij}: V_n(\Gamma_{ij}) \to V_n(\Omega_i)$ 和 $\mathrm{S}_{ij}: V_n(\Gamma_{ij}) \to V_n(\Gamma_{ij})$ 为

$$[\mathrm{R}_{ij}\lambda_{ij}, v]_{\Omega_i} = \langle \lambda_{ij}, v\rangle_{\Gamma_{ij}}, \quad \lambda_{ij} \in V_n(\Gamma_{ij}), \quad \forall v \in V_{hi}(\Omega_i),$$

$$\langle \mathrm{S}_{ij}\lambda_{ij}, \mu_{ij}\rangle = [\mathrm{R}_{ij}\lambda_{ij}, \mathrm{R}_{ij}\mu_{ij}]_{\Omega_i} + [\mathrm{R}_{ji}\lambda_{ij}, \mathrm{R}_{ji}\mu_{ij}]_{\Omega_j},$$

$$\lambda_{ij} \in V_n(\Gamma_{ij}), \forall \mu_{ij} \in V_n(\Gamma_{ij}).$$

令

$$\Gamma'' = \{E \in \{\Gamma_{ij}\} \mid 存在满足\ \partial\Omega_i \cap \partial\Omega = \varnothing\ 的\ \partial\Omega_i,使\ E \subset \partial\Omega_i\}.$$

设 $\Gamma_{ij} \in \Gamma''$,记 $V^0_{ij} = \mathrm{span}\{1\} \subset V_n(\Gamma_{ij})$. 以 $\mathrm{P}^0_{ij}: V_n(\Gamma_{ij}) \to V^0_{ij}$ 表相对内积 $\langle \mathrm{S}_{ij}\cdot, \cdot\rangle_{\Gamma_{ij}}$ 的正交投影算子.对 $\lambda_{ij} \in V_n(\Gamma_{ij})$,记 $\lambda^0_{ij} = \mathrm{P}^0_{ij}, \overline{\lambda}_{ij} = (I - \mathrm{P}^0_{ij}) = \lambda_{ij} - \lambda^0_{ij}$.

引理 10　假定 $\lambda_{ij} \in V_n(\Gamma_{ij})$，则

$$\langle S_{ij}\bar{\lambda}_{ij}, \bar{\lambda}_{ij}\rangle_{\Gamma_{ij}} \leqslant c \mid \lambda_{ij} \mid^2_{-\frac{1}{2},\Gamma_{ij}}, \quad \Gamma_{ij} \in I^v, \tag{4-38}$$

$$\langle S_{ij}\lambda_{ij}, \lambda_{ij}\rangle_{\Gamma_{ij}} \leqslant c \mid \lambda_{ij} \mid^2_{-\frac{1}{2},\Gamma_{ij}}, \quad \Gamma_{ij} \bar{\in} I^v, \tag{4-39}$$

$$\langle S_{ij}\lambda_{ij}, \lambda_{ij}\rangle_{\Gamma_{ij}} \leqslant (\mid D_{ij}\lambda_{ij} \mid^{*2}_{-\frac{1}{2},\partial\Omega_i} + \mid D_{ji}\lambda_{ij} \mid^{*2}_{-\frac{1}{2},\partial\Omega_j}). \tag{4-40}$$

证明　只证(4-40)式，(4-38)和(4-39)式的证明类似．由 $R_{ij}\lambda_{ij}$ 的定义有

$$\begin{aligned}\| R_{ij}\lambda_{ij} \|^{*2}_{1,\Omega_i} &= \langle R_{ij}\lambda_{ij}, \lambda_{ij}\rangle_{\Gamma_{ij}} = \langle R_{ij}\lambda_{ij}, D_{ij}\lambda_{ij}\rangle_{\partial\Omega_i} \\ &\leqslant \| R_{ij}\lambda_{ij} \|^{*}_{\frac{1}{2},\partial\Omega_i} \cdot \mid D_{ij}\lambda_{ij} \mid^{*}_{-\frac{1}{2},\partial\Omega_i} \\ &= \| R_{ij}\lambda_{ij} \|^{*}_{1,\Omega_i} \cdot \mid D_{ij}\lambda_{ij} \mid^{*}_{-\frac{1}{2},\partial\Omega_i},\end{aligned}$$

即

$$\| R_{ij}\lambda_{ij} \|^{*}_{1,\Omega_i} \leqslant \mid D_{ij}\lambda_{ij} \mid^{*}_{-\frac{1}{2},\partial\Omega_i}.$$

同理

$$\| R_{ij}\lambda_{ij} \|^{*}_{1,\Omega_j} \leqslant \mid D_{ij}\lambda_{ij} \mid^{*}_{-\frac{1}{2},\partial\Omega_j}.$$

因此，由 $S_{ij}\lambda_{ij}$ 的定义即知(4-40)式成立．

现给出定理 3 的证明．定义 $V_n(\Gamma)$ 的粗子空间为

$$V_0 = \mathrm{span}\{D_{ij}1 \mid \Gamma_{ij} \subset \partial\Omega_i，对某个满足\ \partial\Omega_i \cap \partial\Omega = \varnothing\ 的\ i\}，$$

以 $S_0: V_0 \to V_0$ 表 S 的限制算子(见第 2 章)，$Q_0: V_n(\Gamma) \to V_0$ 表 L^2 投影算子，可直接验证 $\bar{M}_1$ 和 $\bar{M}_2$ 的算子形式分别为

$$M_1 = \sum_{i<j} D_{ij}S_{ij}^{-1}D_{ij}^T, \quad M_2 = S_0^{-1}Q_0 + M_1.$$

仅考虑定理 3(1°)，定理 3(2°)的证明类似．由第 2 章的一般理论知，要证定理 3(1°)，只需证(相当于 $V_0 = \{0\}$)：

(1) 对任意 $\lambda_{ij} \in V_n(\Gamma_{ij})$，有

$$\langle S(\sum_{i<j} D_{ij}\lambda_{ij}), \quad \sum_{i<j} D_{ij}\lambda_{ij}\rangle \leqslant C\sum_{i<j}\langle S_{ij}\lambda_{ij}, \lambda_{ij}\rangle_{\Gamma_{ij}}. \tag{4-41}$$

(2) 对任意 $\lambda \in V_n(\Gamma)$，有

$$\sum_{i<j}\langle S_{ij}\lambda_{ij}, \lambda_{ij}\rangle_{\Gamma_{ij}} \leqslant C(1+\log^2 n)\langle S\lambda, \lambda\rangle, \tag{4-42}$$

其中 $\lambda_{ij} = \lambda \mid_{\Gamma_{ij}}$.

(4-41)式的证明是标准的(参见文献[14])，下证(4-42)式．

事实上，由(4-40)，(4-33)和(4-25)式得

$$\begin{aligned}\langle S_{ij}\lambda_{ij}, \lambda_{ij}\rangle_{\Gamma_{ij}} &\leqslant (\mid D_{ij}\lambda_{ij} \mid^{*}_{-\frac{1}{2},\partial\Omega_i} + \mid D_{ji}\lambda_{ij} \mid^{*}_{-\frac{1}{2},\partial\Omega_j}) \\ &\leqslant C(1+\log^2 n)(\mid \lambda_i \mid^{*2}_{-\frac{1}{2},\partial\Omega_i} + \mid \lambda_j \mid^{*2}_{-\frac{1}{2},\partial\Omega_j}) \\ &\leqslant C(1+\log^2 n)(\langle S_i\lambda, \lambda\rangle + \langle S_j\lambda, \lambda\rangle),\end{aligned}$$

其中 $\lambda_i = \lambda \mid_{\partial\Omega_i}, \lambda_j = \lambda \mid_{\partial\Omega_j}$. 由此即推得(4-42)式．

参 考 文 献

1 Bernardi C, Moday Y, Patera A. A *new nonconforming approach to domain decomposition*: the mortar element method. In: Brezis H Lions J L eds. Nonlinear partial differential equations and their applications. Pitman, 1989.
2 Bramble J H, Pasciak J, Schatz A. The construction of preconditions for elliptic problems by substructuring. I. Math Comp, 1986(47), 103 ~ 134.
3 Bramble J H, Pasciak J, Schatz A. The construction of preconditioners for elliptic problems by substructuring. IV. Math Comp, 1989(53): 1 ~ 24.
4 蔡大用,白峰杉.高等数值分析.北京:清华大学出版社.1997.
5 Chan T F, Mathew T. Domain decomposition algorithms. Acta Numerià, 61 ~ 143.
6 Dinh Q V, Glowinski R, Periaux J. Solving elliptic problems by domain decom-position methods with applications. in elliptic problem solver Ⅱ. New York: Academic Press, 1982.
7 Dryja M. A capacitance matrix method for dirilhlet problems on polygon region. Numer Math, 1982(39): 51 ~ 64.
8 Dryja M A. Method of domain decomposition for 3-D finite elment problems. In: Glowinski R, Golub G, Meurant G et al eds. First International Symposium on Domain Decomposition Methods for PDEs. Philadelphia: SIAM, 1988.
9 Dryja M., Widlund O. Domain decomposition algorithms with small overlap. SIAM J Sci Comput, 1994(3): 604
10 Farhat C. A saddle-point principle domain decomposition method for the solution of solid methanics problems. In: David E, Keys, Maurant G et al eds. 5th International Symposium on Domain Decomposition Methods for Partial Differential Equations. Philadelphia: SIAM, 1992.
11 Farhat C, Mandel J, Chen Poshu. A scalalble lagrange multiphior based domain decomposition method for time-dependent problems. Int J Numer Methods Eng, 1995(38): 3831 ~ 3853.
12 Glowinski R. Numerical methods for nonlinear variational prblems. Springer Series in Comparative Physics. New York: Springer-Verlag, 1984.
13 顾金生,胡显承.基于子结构法构造用非协调元解椭圆型问题的预处理器(Ⅰ).计算数学,1996,18(2):114 ~ 128.
14 胡齐芽.非匹配网格区域分解方法研究:[博士论文].北京:中科院数学所,1998.
15 胡齐芽,梁国平.界面预条件子构造的一般框架,计算数学,1999,21(1)
16 康立山等.解数学物理问题的异步并行算法.北京:科学出版社,1985.
17 梁国平,梁平.杂交有限元的区域分解法.计算数学,1989(11):323 ~ 332.
18 梁国平,何江衡.非协调区域分解的 Lagrangian 乘子法.计算数学,1992(14):207

~215.

19 Lions P L. On the Schwarz alternating method Ⅲ: a variant for Nonoverlapping subdomains. In: T Chan, R Glowinski, J Periaux et al eds. Third International Symposium on Domain Decomposition Methods for PDEs. Philadelphia: SIAM, 1990.

20 吕涛,石济民,林振宝.区域分解算法——偏微分方程数值解新技术.北京:科学出版社,1992.

21 Mandel J. Balancing domain decomposition. Commun Numer Meth Engrg, 1993(9): 233~241.

22 Mc Cormick S. Fast adaptive composite grid(FAC) methods: theory for the variational case, in defect correction methods. Theory and Applications Computations Supplementation. 1984(5): 115~122.

23 Mc Cormick S. Multilevel adaptive methods for partial differential equations. Philadephia: SIAM, 1989.

24 米赫林.二次泛函的极小问题.王维新译.北京:科学出版社,1964.

25 Miller K. Numerical analogs to the Schwarz alternating procedure. Numer Math, 1965(7): 91~103.

26 Proskurowski W, Widlund O. On the numerical solution of helmholtz's equation by the capacitance matrix method math. Comput, 1976(30): 433~468.

27 Schwarz H. Gesammelete mathematische abhandlungen. Vol 2. Berlin: Springer, 1890.

28 Shi Z, Xie Z, Substructure preconditioners for nonconforming plate element. J Comp Math, 1998(4): 289~304.

29 Smith B, Bjrstad P, Gropp W. Domain decomposition. London: Cambridge University Press, 1996.

30 孙澎涛.依赖时间问题的 Lagrange 乘子非协调区域分解法:[博士论文].北京:中科院数学所,1997.

31 Tallec P. Le domain decomposition methods in computational mechanies. Comput Mech Adv, 1994(2): 121~220.

32 Xu J, Zou J. Some nonoverlapping domain decomposition methods. SIAM Review, 1998(4).

33 余德浩.无界区域非重叠区域分解算法的离散化及其收敛性.计算数学,1995,18(3): 328~336.

34 余德浩.自然边界元方法的数学理论.北京:科学出版社,1993 年.

·计算机数学卷·

第7篇

小波分析

编　者　刘智新
审校者　曲秉玉

目　录

引　言

小波分析是近年来发展起来的理论与应用的数学分支．主要用来分析、处理奇异性强的不平稳信号．由于传统的傅里叶分析方法对于奇异信号处理效果较差，因而小波分析已被广泛地应用于图像处理、语言合成、边缘探测、石油勘探等许多方面，成为处理突变信号及不平稳信号的重要工具．小波这一名称首先是由法国地质学家 J. Morlet 与 A. Grossmann 在分析地质数据时引进的．Y. Meyer, S. Mallat 及 I. Daubechies 等人都对于小波理论的发展作出了重要的贡献．目前小波分析仍是国际上活跃的研究领域之一．

1　小波分析基础

1.1　信号的时频局部化分析

1.1.1　信号及其时频局部化

对于给定的连续信号 $f(t)$，它在频域上可以表征为 $f(t)$ 的傅里叶(Fourier)变换：

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt. \qquad (1\text{-}1)$$

(1-1)式给出的是信号 $f(t)$ 在整个时域上频谱的性质．然而在实际问题中，人们常需要知道 $f(t)$ 在某一时间段 $[t, t+\Delta t]$ 中的局部频谱特征．这就是信号的时频局部化问题．例如地质信号，人们关心的是在什么位置出现"断层"，即信号在什么位置出现频率奇异性．对于这一类奇异性强的不平稳信号，传统的傅里叶变换方法常常效果不佳．

1.1.2　波函数的时频局部化

分析产生上述问题的主要原因，可以发现，傅里叶变换(1-1)中的波函数 e^{it} 虽然在频域上具有最好的局部化性质，但在时域上却不具有任何局部化特性．因而需要寻找另外的波函数 $g(t)$ 来代替 e^{it}，使之能同时反映出信号的时间域及频率域的局部化特征．这就是说波函数 $g(t)$ 应具备时域及频域的局部化特性．

1.1.3　波函数时频局部化的表征

根据海森伯格(Heisenberg)测不准原理，任何波函数在时域及频域同时局部化，在严格意义下是不可能的，但是在概率意义下则是可能的．

1. 时间中心与频率中心

设 $g \in L^2(\mathbf{R})$ 且 $tg(t) \in L^2(\mathbf{R})$.

$$t_0 = \int_{-\infty}^{\infty} t \mid g(t) \mid^2 \mathrm{d}t \Big/ \int_{-\infty}^{\infty} \mid g(t) \mid^2 \mathrm{d}t,$$

$$\omega_0 = \int_{-\infty}^{\infty} \omega \mid \hat{g}(\omega) \mid^2 \mathrm{d}\omega \Big/ \int_{-\infty}^{\infty} \mid \hat{g}(\omega) \mid^2 \mathrm{d}\omega$$

分别称为 $g(t)$ 的时间中心与频率中心．在相平面上，(t_0, ω_0) 称为 $g(t)$ 的窗口中心．

2. 时间宽度与频率宽度

$$\Delta_g = \left[\int_{-\infty}^{\infty} (t - t_0)^2 \mid g(t) \mid^2 \mathrm{d}t \Big/ \int_{-\infty}^{\infty} \mid g(t) \mid^2 \mathrm{d}t\right]^{1/2}$$

与
$$\Delta_{\hat{g}} = \left[\int_{-\infty}^{+\infty} (\omega - \omega_0)^2 \mid \hat{g}(\omega) \mid^2 \mathrm{d}\omega \Big/ \int_{-\infty}^{+\infty} \mid \hat{g}(\omega) \mid^2 \mathrm{d}\omega\right]^{1/2}$$

分别称为 $g(t)$ 的时间宽度与频率宽度．

显然，$g(t)$ 的能量主要集中在区间$(t_0 - \Delta_g, t_0 + \Delta_g)$内，$g(t)$ 的主要频谱集中在区间$(\omega_0 - \Delta_{\hat{g}}, \omega_0 + \Delta_{\hat{g}})$内．$\Delta_g$ 与 $\Delta_{\hat{g}}$ 的大小刻划了波函数 $g(t)$ 在时域及频域的局部化程度．

3. 海森伯格测不准原理

对任何 $g(t) \in L^2(\mathbf{R})$，若 $tg(t)$ 及 $\omega\hat{g}(\omega)$ 都属于 $L^2(\mathbf{R})$，则，

$$\Delta_g \Delta_{\hat{g}} \geqslant \frac{1}{2}.$$

1.2 连续小波变换

1.2.1 连续小波变换的概念

为了克服傅里叶变换不能时、频同时局部化的缺点，选取满足下列条件的函数 $\psi(t)$ 作为波函数，称为基本小波：

1° $\psi \in L^2(\mathbf{R})$；

2° $C_\psi = 2\pi \int_{\mathbf{R}} \mid \hat{\psi}(\omega) \mid^2 \mathrm{d}\omega / \mid \omega \mid < \infty$. (1-2)

条件2°称为基本小波 $\psi(t)$ 的容许条件．同时为兼顾整个时域及频域，需要考虑由 $\psi(t)$ 得到的一族波函数，

$$\psi_{ab}(t) = \mid a \mid^{-1/2} \psi\left(\frac{t - b}{a}\right),$$

其中 a、$b \in \mathbf{R}$ 且 $a \neq 0$. 对于任何信号 $f(t) \in L^2(\mathbf{R})$，定义 $f(t)$ 的连续小波变换为

$$\begin{aligned} W_f(a, b) &= \langle f, \psi_{ab} \rangle \\ &= \mid a \mid^{-1/2} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t - b}{a}\right)} \mathrm{d}t. \end{aligned} \quad (1\text{-}3)$$

1.2.2　信号的恢复

由信号 $f(t)$ 的小波变换可以恢复出 $f(t)$[①]

$$f(t) = C_\psi^{-1}\int_{-\infty}^{\infty}\frac{\mathrm{d}a}{a^2}\int_{-\infty}^{\infty}\psi_{ab}(t)W_f(a,b)\mathrm{d}b, \tag{1-4}$$

其中 C_ψ 由(1-2)式定义．

例 1　设 $\mathrm{supp}\widehat{f}$、$\mathrm{supp}\widehat{\psi}\subseteq[0,+\infty)$，则当 $a<0$ 时，

$$\begin{aligned}W_f(a,b) &= \int_{-\infty}^{\infty}f(t)\,|a|^{-1/2}\,\overline{\psi\left(\frac{t-b}{a}\right)}\mathrm{d}t\\ &= \frac{1}{2\pi}\,|a|^{1/2}\int_{-\infty}^{\infty}\widehat{f}(\omega)\,\overline{\widehat{\psi}(a\omega)\mathrm{e}^{-\mathrm{i}b\omega}}\mathrm{d}\omega = 0,\end{aligned}$$

从而此时

$$f(t) = C_\psi^{-1}\int_{0}^{\infty}\frac{\mathrm{d}a}{a^2}\int_{-\infty}^{\infty}W_f(a,b)\psi_{ab}(t)\mathrm{d}b.$$

1.2.3　小波变换的时频局部化特性

为方便起见，通常对基本小波附加一组限制条件：

1° $\widehat{\psi}(0) = \dfrac{1}{\sqrt{2\pi}}\displaystyle\int_{-\infty}^{\infty}\psi(t)\mathrm{d}t = 0$；

2° $|\psi(t)|\leqslant C(1+|t|)^{-(1+\epsilon)}$，$t\in\mathbf{R}$（$\epsilon>0$ 为常数）；

3° $\|\psi\|_2 = 1$；

4° $\displaystyle\int_{-\infty}^{+\infty}t\,|\psi(t)|^2\mathrm{d}t = 0$.

1. 小波函数的窗口

对于小波函数 $\psi_{ab}(t)$，它的时间中心与正、负频率中心分别为

$$t_{ab} = \int_{-\infty}^{\infty}t\,|\psi_{ab}(t)|^2\mathrm{d}t,$$

$$\omega_{ab}^{\pm} = \int_{0<\pm\omega<\infty}\omega\,|\widehat{\psi}_{ab}(\omega)|^2\mathrm{d}\omega;$$

它的时间宽度与正、负频率宽度分别为

$$\Delta_{\psi_{ab}} = \left(\int_{-\infty}^{\infty}(t-t_{ab})^2\,|\psi_{ab}(t)|^2\mathrm{d}t\right)^{1/2},$$

$$\Delta_{\widehat{\psi}_{ab}}^{\pm} = \left(\int_{0<\pm\omega<\infty}(\omega-\omega_{ab}^{\pm})^2\,|\widehat{\psi}_{ab}(\omega)|^2\mathrm{d}\omega\right)^{1/2}.$$

2. 小波变换的"变焦距"性质

不难直接算得

---

① 公式(1-4)在下述意义下成立：

$$\lim_{\substack{A,B\to+\infty\\ \epsilon\to 0^+}}\left\|f - C_\psi^{-1}\iint_{\substack{\epsilon<|a|<A\\ |b|<B}}W_f(a,b)\psi_{ab}(t)\frac{\mathrm{d}a\mathrm{d}b}{a^2}\right\|_2 = 0.$$ 详见文献[2].

$$t_{ab} = b; \tag{1-5}$$
$$\omega_{ab}^{\pm} = \omega^{\pm}/a; \tag{1-6}$$
$$\Delta_{\psi_{ab}} = a\Delta_{\psi}; \tag{1-7}$$
$$\Delta_{\hat{\psi}_{ab}}^{\pm} = \frac{1}{a}\Delta_{\hat{\psi}}^{\pm}. \tag{1-8}$$

其中

$$\omega^{\pm} = \int_{0<\pm\omega<\infty} \omega \mid \hat{\psi}(\omega) \mid^2 d\omega;$$
$$\Delta_{\psi} = \left(\int_{-\infty}^{+\infty} t^2 \mid \psi(t) \mid^2 dt\right)^{1/2};$$
$$\Delta_{\hat{\psi}}^{\pm} = \left(\int_{0<\pm\omega<\infty} (\omega - \omega^{\pm})^2 \mid \hat{\psi}(\omega) \mid^2 d\omega\right)^{1/2};$$

分别为 ψ 的正、负频率中心、时间宽度及正、负频率宽度．由此可得出如下结论：

(1) 根据(1-5)及(1-6)式，小波函数 $\psi_{ab}(t)$ 的一对窗口中心位于相平面上$(b, \omega^{\pm}/a)$处，并且当参数 a、b 变动时，$(b, \omega^{\pm}/a)$ 可以在相平面上任何位置．

(2) 根据(1-7)及(1-8)式，小波函数 $\psi_{ab}(t)$ 的确定窗口面积对于任意的 a、b 是不变的，

$$2\Delta_{\psi_{ab}} \cdot 2\Delta_{\hat{\psi}_{ab}}^{\pm} = 4\Delta_{\psi}\Delta_{\hat{\psi}}^{\pm}.$$

然而，当 a、b 变动时，窗口的形状却是不同的．$1/a$ 越大，窗口的时间宽度越小，同时频率宽度越大．这表明，小波变换(1-3)的时频局部化程度由时域参数 b 及频域参数 $1/a$ 决定．在频率较高的频段(即 $1/a$ 较大时)，时间域的局部化程度较高，即分辨率较高．这就是说小波变换具有“变焦距”的特性．

1.2.4 用小波变换来刻划信号的局部正则性

定理1 设基本小波 $\psi(t)$ 满足 $\hat{\psi}(0) = 0$，且

$$\int_{-\infty}^{\infty} (1 + \mid t \mid) \mid \psi(t) \mid dt < \infty.$$

若 $f(t) \in \mathrm{Lip}\alpha, 0 < \alpha \leqslant 1$，即存在常数 C 使得

$$\mid f(t) - f(u) \mid \leqslant C \mid t - u \mid^{\alpha}, \quad \forall t, u \in \mathbf{R},$$

那么

$$\mid W_f(a,b) \mid \leqslant C_1 \mid a \mid^{\alpha+\frac{1}{2}},$$

其中 C_1 为常数．

反之，若再设 $\mathrm{supp}\psi$ 紧，$f(t)$ 为 $L^2(\mathbf{R})$ 中连续有界函数，且存在 $\alpha \in (0,1)$ 及常数 C_2，使得

$$\mid W_f(a,b) \mid \leqslant C_2 \mid a \mid^{\alpha+1/2},$$

那么存在常数 C_3，使得 $f \in \mathrm{Lip}_{C_3}\alpha$.

1.2.5 连续小波变换的推广形式

连续小波变换(2-2)可以作如下推广：在变换与恢复时使用不同的基本小波

函数.

设 ψ、$\tilde{\psi} \in L^2(\mathbf{R})$ 且满足

$$\int_{-\infty}^{\infty} |\hat{\psi}(\omega)| |\hat{\tilde{\psi}}(\omega)| \frac{d\omega}{|\omega|} < \infty.$$

若 $W_f(a,b)$ 由(2-2)式给出,那么

$$f(t) = C_{\psi\tilde{\psi}}^{-1}\int_{-\infty}^{\infty} \frac{da}{a^2}\int_{-\infty}^{\infty} W_f(a,b)\tilde{\psi}_{ab}(t)db,$$

其中

$$C_{\psi\tilde{\psi}} = \int_{-\infty}^{\infty} 2\pi\overline{\hat{\psi}(\omega)}\hat{\tilde{\psi}}(\omega)|\omega|^{-1}d\omega \neq 0.$$

进步的讨论参见文献[2].

2 离散小波变换与小波框架

2.1 离散小波变换

2.1.1 连续小波的离散化

在傅里叶分析中,傅里叶级数可以看作是傅里叶变换的离散化.考虑连续小波变换的离散化,也就是把小波函数 $\psi_{ab}(t) = |a|^{-1/2}\psi(\frac{t-b}{a})$ 中的参数 a、b 离散化.若选取基本小波函数 $\psi(t)$ 具有"较好"的性质,可以只考虑 $a>0$的情形.取 a 分别为 a_0^m,b 分别为 $nb_0a_0^m$,m、$n = 0, \pm 1, \pm 2, \cdots$ 其中 $a_0 > 1$,$b_0 > 0$为常数,便得到函数系

$$\psi_{mn}(t) = a_0^{m/2}\psi(a_0^m t - b_0 n), \quad m,n = 0, \pm 1, \pm 2\cdots. \tag{2-1}$$

通常取最简单的情形:$a_0 = 2, b_0 = 1$.

2.1.2 离散小波变换的基本问题

对于(2-1)中给出的函数系$\{\psi_{mn}\}_{m,n}$,任取$f \in L^2(\mathbf{R})$,即可得到$f(t)$的离散小波变换系数列$\{\langle f,\psi_{mn}\rangle\}_{m,n=-\infty}^{+\infty}$.

需要考虑的基本问题是:

(1) 是否对任何$f(t) \in L^2(\mathbf{R})$,系数列$\{\langle f,\psi_{mn}\rangle\}_{mn}$都可以完全地刻划$f$,即当$f$改变时,对应的系数列$\{\langle f,\psi_{mn}\rangle\}$是否一定会改变.

(2) 更进一步地,是否可以构造出数值上稳定的方法,使得对于任何的$f(t) \in L^2(\mathbf{R})$,从系数列$\{\langle f,\psi_{mn}\rangle\}_{mn}$重构$f(t)$?

(3) 是否每个$f(t) \in L^2(\mathbf{R})$都可以写成$\{\psi_{mn}\}_{mn}$的级数形式(即小波级数)?也

就是说,是否存在数列$\{C_{mn}\}$,使得

$$f(t)=\sum_{mn}C_{mn}\psi_{mn}(t).$$

若存在,可否找出数值上稳定的方法求出$\{C_{mn}\}$?

这里需要指出,函数系$\{\psi_{mn}\}$是否线性无关,在一些问题中并不重要,甚至在有些情况下可能带来方便．但是数值计算上的稳定性要求却是关键的．在下一节中将详细讨论这一问题．

2.2　框　　架

2.2.1　框架的概念

为了研究离散小波变换数值计算上的稳定性,引入框架的概念．

希尔伯特空间H中的向量族$\{\varphi_j\}$称为一个框架,若存在正数A、B使得对任何$f\in H$有

$$A\|f\|_2^2\leqslant\sum_j|\langle f,\varphi_j\rangle|^2\leqslant B\|f\|_2^2. \tag{2-2}$$

A、B称为框架的上、下界(或框架常数).

当$A=B$时,称$\{\varphi_j\}$为H中的紧框架．

关于框架与正交基的关系,有如下结论:

若$\{\varphi_j\}$为H中的框架,$\|\varphi_j\|=1$,且框架常数$A=B=1$,那么$\{\varphi_j\}$为H中的标准正交基．

现在可以对2.1.2中提出的问题(1)给出肯定的回答,这只需要使得$\{\psi_{mn}(t)\}$成为$L^2(\mathbf{R})$的框架即可．

2.2.2　对偶框架

为了回答2.1.2中的问题(2),对于H中的框架$\{\varphi_j\}$定义H上的算子T如下:

$$Tf=\sum_j\langle f,\varphi_j\rangle\varphi_j,\quad f\in H.$$

(1) 由于$\langle Tf,f\rangle=\sum_j\langle f,\varphi_j\rangle\langle\varphi_j,f\rangle=\sum_j|\langle f,\varphi_j\rangle|^2$,根据框架性质(2-2)式,有

$$A\|f\|_2^2\leqslant\langle Tf,f\rangle\leqslant B\|f\|_2^2,\quad\forall f\in H,$$

从而不难验证T为H上的有界正线性算子,且

$$A\leqslant\|T\|\leqslant B.$$

(2) 由(1)进而可得T在H上的逆算子存在并且

$$B^{-1}\leqslant\|T^{-1}\|\leqslant A^{-1}.$$

(3) 根据T的定义,有

$$f=T^{-1}[Tf]=T^{-1}(\sum_j\langle f,\varphi_j\rangle\varphi_j)$$
$$=\sum_j\langle f,\varphi_j\rangle T^{-1}\varphi_j.$$

若记
$$\tilde{\varphi}_j = T^{-1}\varphi_j, \quad \forall j, \tag{2-3}$$
则有
$$f = \sum_j \langle f, \varphi_j \rangle \tilde{\varphi}_j, \quad \forall f \in H. \tag{2-4}$$

(4) 对于向量系$\{\tilde{\varphi}_j\}$,可以证明(见文献[3],命题1.2.10.)

定理1　由(2-3)式定义的向量族$\{\tilde{\varphi}_j\}$亦为H中的框架,其框架上、下界分别为A^{-1}与B^{-1}.

一般地,如果给定框架$\{\varphi_j\}$和$\{\varphi_j\}$满足(2-4)式,则称$\{\tilde{\varphi}_j\}$为$\{\varphi_j\}$的**对偶框架**.

例1　在$\mathbf{R}^2$中取三个向量$e_1 = (0,1), e_2 = (-\sqrt{3}/2, -1/2), e_3 = (\sqrt{3}/2, -1/2)$. 考虑由$e_1, e_2, e_3$构成的向量系. 对任何$v = (t_1, t_2) \in \mathbf{R}^2$,容易算出,
$$\sum_{j=1}^{3} |\langle v, e_j \rangle|^2 = \frac{3}{2}(t_1^2 + t_2^2) = \frac{3}{2}\langle v, v \rangle.$$
所以$\{e_1, e_2, e_3\}$构成了$\mathbf{R}^2$的紧框架.

注意到$\{e_1, e_2, e_3\}$在$\mathbf{R}^2$中不是线性独立的,框架常数$A = B = 3/2$说明$\{e_1, e_2, e_3\}$较之$\mathbf{R}^2$的基"过度"地重叠之程度.

通过简单计算可得,
$$v = \frac{2}{3}\sum_{j=1}^{3}\langle v, e_j \rangle e_j.$$
所以$\tilde{e}_j = 2/3 e_j, j = 1,2,3$,即为$\{e_1, e_2, e_3\}$的一个对偶框架,其框架常数为$A^{-1} = B^{-1} = 2/3$.

2.2.3　对偶框架的性质

(1) 框架$\{\varphi_j\}$与其对偶框架$\{\tilde{\varphi}_j\}$的关系是相互的. 因为对于任何f、$g \in H$,
$$\langle f, g \rangle = \langle \sum_j \langle f, \varphi_j \rangle \tilde{\varphi}_j, g \rangle = \sum_j \langle f, \varphi_j \rangle \langle \tilde{\varphi}_j, g \rangle,$$
同理
$$\langle g, f \rangle = \sum_j \langle g, \tilde{\varphi}_j \rangle \langle \varphi_j, f \rangle,$$
所以
$$g = \sum_j \langle g, \tilde{\varphi}_j \rangle \varphi_j,$$
即$\{\varphi_j\}$亦为$\{\tilde{\varphi}_j\}$的对偶框架.

(2) 框架$\{\varphi_j\}$通常不一定线性独立,因而每个向量f用$\{\varphi_j\}$的表示也不一定唯一. 但是下面结论说明使用由对偶框架生成的系数列$\{\langle f, \tilde{\varphi}_j \rangle\}$作为分解系数为"最佳"表示(见文献[2],命题3.2.4).

定理2　设$\{\varphi_j\}$为希尔伯特空间H的框架,$\{\tilde{\varphi}_j\}$为它的对偶框架,又设$f = \sum_j c_j\varphi_j$,使得$\sum_j |c_j|^2 < +\infty$,那么

$$\sum_j |c_j|^2 \geqslant \sum_j |\langle f, \tilde{\varphi}_j \rangle|^2,$$

等号当且仅当所有的 $c_j = \langle f, \tilde{\varphi}_j \rangle$ 时成立．

(3) 若 $\{\varphi_j\}$ 为线性独立的框架，则其对偶框架是唯一的，并且 $\{\varphi_j\}$ 与其对偶框架双正交：

$$\langle \varphi_j, \tilde{\varphi}_k \rangle = \delta_{jk} = \begin{cases} 0, & k \neq j, \\ 1, & k = j. \end{cases}$$

反之，若框架 $\{\varphi_j\}$ 与其对偶框架双正交，那么 $\{\varphi_j\}$ 线性独立．

2.2.4　框架与里斯基

希尔伯特空间 H 中向量族 $\{\varphi_j\}$ 称为 H 的里斯(Riesz)基，若对任何 $f \in H$，f 可表示为 $f = \sum_j c_j\varphi_j$，并且存在与 $f \in H$ 无关的正数 A、B，使得

$$A\sum_j |c_j|^2 \leqslant \|f\|_2^2 \leqslant B\sum_j |c_j|^2. \tag{2-5}$$

不难验证，里斯基一定是线性独立的．对于里斯基与框架的关系，有

定理3　希尔伯特空间中向量族 $\{\varphi_j\}$ 为 H 的里斯基，当且仅当它为 H 中的线性独立框架．此时(2-5)式中的 A、B 即为 $\{\varphi_j\}$ 的框架常数．

2.3　小波框架

2.3.1　小波框架的概念

设 $\psi(t)$ 为一个基本小波函数．若(2-1)式中定义的函数系 $\{\psi_{mn}\}$ 构成 $L^2(\mathbf{R})$ 中的框架，则称它为 $L^2(\mathbf{R})$ 的一个小波框架．

2.3.2　小波框架的生成条件

1. 一个必要条件

设 $\psi(t)$ 为基本小波函数，$\{\psi_{mn}\}$ 如(2-1)式．若 $\{\psi_{mn}\}$ 构成 $L^2(\mathbf{R})$ 的框架，以 A 与 B 为框架的下界与上界，那么

$$\frac{1}{2\pi}Ab_0\ln a_0 \leqslant \int_{0<\pm\omega<\infty} |\hat{\psi}(\omega)|^2 \frac{d\omega}{|\omega|} \leqslant \frac{1}{2\pi}Bb_0\ln a_0.$$

2. 一个充分条件

设 $\psi(t)$ 与 $\{\psi_{mn}\}$ 如上，若存在正数 $\alpha > 0, \gamma > \alpha + 1$ 及 $c > 0$，使得

$$|\hat{\psi}(\omega)| \leqslant c|\omega|^{\alpha}(1+|\omega|)^{-\gamma}, \quad \omega \in \mathbf{R},$$

那么存在正数 d，使得对每个 $a_0 > 1$ 及 $b_0 \in (0, d)$，函数系 $\{\psi_{mn}\}$ 都构成 $L^2(\mathbf{R})$ 的小波框架．

更详细的讨论见文献[2]．

2.3.3 关于小波框架的对偶框架

这里仅指出以下几点：

(1) 当$\{\psi_{mn}\}$为小波框架时，可以通过2.2.2中定义的算子T及T^{-1}得到它的对偶框架．这在某些情况下，在数值上是可行的．尤其当A与B非常接近或$A=B$的情形．

(2) 在某些情况下，$\{\psi_{mn}\}$的对偶框架$\{\tilde{\psi}_{mn}\}$可以由一个函数$\tilde{\psi}$依(2-1)式的方式产生：

$$\tilde{\psi}_{mn}(t)=a_0^{m/2}\tilde{\psi}(a_0^m t-b_0 n).$$

此时称$\{\psi_{mn}\}$与$\{\tilde{\psi}_{mn}\}$互为对偶小波框架．

特别地，若$\{\psi_{mn}\}$还是线性独立的，那么$\{\psi_{mn}\}$与$\{\tilde{\psi}_{mn}\}$还是双正交的．称它们为**双正交小波基**．

(3) 若$\{\psi_{mn}\}$构成$L^2(\mathbf{R})$的标准正交基，那么$\tilde{\psi}=\psi$，且$\forall f\in L^2(\mathbf{R})$，

$$f(t)=\sum_{jk}\langle f,\psi_{jk}\rangle\psi_{jk}(t).$$

此时称$\{\psi_{mn}\}$为**正交小波基**，$\psi(t)$为**正交小波**．

3 多尺度分析与正交小波基

3.1 多尺度分析与正交小波基的构造

3.1.1 多尺度分析的概念

S. Mallat和Y. Meyer于1986年引入了多尺度分析的概念，为正交小波基的构造提供了一个一般的方法．

定义1 $L^2(\mathbf{R})$的一个多尺度分析，是$L^2(\mathbf{R})$的一个闭子空间列$\{V_j\}_{-\infty}^{\infty}$与$V_0$中一个函数$\varphi$的联合体，且满足下列条件：

1° $V_j\subseteq V_{j+1}, j=0,\pm1,\pm2\cdots$

2° $\bigcap_j V_j=\{0\}, \overline{\bigcup_j V_j}=L^2(\mathbf{R})$;

3° $f(t)\in V_j$，当且仅当$f(2t)\in V_{j+1}$;

4° $\{\varphi(t-n)\}_{n=-\infty}^{+\infty}$构成$V_0$的标准正交基．函数$\varphi(t)$称为**尺度函数**．

由定义1-1易得：

(1) $f(t)\in V_0$，当且仅当$g(t)=f(2^j t)\in V_j$.

(2) 记

$$\varphi_{mn}(t) = \sqrt{2}^m \varphi(2^m t - n), \quad m, n = 0, \pm 1 \cdots \tag{3-1}$$

(即在(2-1)式中取 $a_0 = 2, b_0 = 1$,以下总作此假定)则$\{\varphi_{mn}\}_{n=-\infty}^{+\infty}$为$V_m$的标准正交基,当且仅当$\{\varphi(t-n)\}_n$为$V_0$的标准正交基.

3.1.2　尺度函数与双尺度方程

既然尺度函数 $\varphi(t) \in V_0 \subseteq V_1$,而$\{\varphi_{1n}\}_n$为$V_1$的标准正交基,从而存在数列$\{h_n\}_{n=-\infty}^{\infty}$使得

$$\varphi(t) = \sqrt{2}\sum_n h_n \varphi(2t - n). \tag{3-2}$$

(3-2)式称为**双尺度方程**,它是小波基构造中的一个基本方程.

考察方程(3-2)两端的傅里叶变换,可得

$$\hat{\varphi}(\omega) = m_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right), \quad \omega \in \mathbf{R}.$$

即

$$\hat{\varphi}(2\omega) = m_0(\omega)\hat{\varphi}(\omega), \tag{3-3}$$

其中

$$m_0(\omega) = \sqrt{2}^{-1}\sum_n h_n e^{-in\omega}. \tag{3-4}$$

通常称 $m_0(\omega)$ 为(关于尺度函数 $\varphi(t)$ 的)**滤波函数**.

为进一步研究尺度函数,给出下列命题.

命题1　设 $g(t) \in L^2(\mathbf{R})$,则$\{g(t-n)\}_{n=-\infty}^{+\infty}$构成它所张成的线性空间$\overline{\mathrm{span}\{g(t-n) \mid n \in \mathbf{Z}\}}$的标准正交基,当且仅当

$$\sum_{k=-\infty}^{+\infty} |\hat{g}(\omega + 2k\pi)|^2 \equiv 1/2\pi, \quad \omega \in \mathbf{R}.$$

根据命题3-1,不难从(3-3)式中得到

$$|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 \equiv 1, \quad \omega \in \mathbf{R}. \tag{3-5}$$

下列命题给出了关于尺度函数 $\varphi(t)$ 及滤波函数 $m_0(\omega)$ 的几个基本事实.它可由(3-3)及(3-5)式导出.

命题2　设尺度函数 $\varphi(t)$ 的傅里叶变换 $\hat{\varphi}(\omega)$ 在 $\omega = 0$ 附近连续,则有

1° $\hat{\varphi}(0) \neq 0$;

2° $m_0(0) = 1, m_0(\pi) = 0$;

3° $\hat{\varphi}(2k\pi) = 0$,当 $k \neq 0$ 时;

4° $|\hat{\varphi}(0)| = 1$.

依此命题,以下总假定:

$$\hat{\varphi}(0) = 1. \tag{3-6}$$

3.1.3　基本小波与正交小波基的构造

定理1　设$(\{V_j\}, \varphi)$为$L^2(\mathbf{R})$的一个多尺度分析,令

$$\psi(t) = \sqrt{2}\sum_{n=-\infty}^{\infty}(-1)^n \bar{h}_{1-n}\varphi(2t-n), \tag{3-7}$$

其中$\{h_n\}$由(3-2)式确定．又记

$$\psi_{mn}(t) = \sqrt{2}^m\psi(2^m t - n), \quad n = 0, \pm 1, \pm 2\cdots$$

那么$\{\psi_{mn}\}_{mn}$为$L^2(\mathbf{R})$的一个正交小波基．若记W_j为V_j在V_{j+1}中的正交补，即

$$V_{j+1} = V_j \oplus W_j, \quad j = 0, \pm 1, \pm 2\cdots$$

$$\overline{\bigoplus_{j=-\infty}^{\infty} W_j} = \overline{\bigcap_{j=-\infty}^{\infty} V_j} = L^2(\mathbf{R}),$$

那么$\{\psi_{mn}\}_{n=-\infty}^{+\infty}$为$W_m$的标准正交基．

定理1给出了由多尺度分析来构造正交小波基的一般方法．(3-7)式给出了正交小波函数$\psi(t)$．

它的另一表述为

$$\hat{\psi}(2\omega) = m_1(\omega)\hat{\varphi}(\omega), \tag{3-8}$$

其中$m_1(\omega)$为对应于$\hat{\psi}(t)$的滤波函数，

$$m_1(\omega) = \sqrt{2}^{-1}\sum_n(-1)^n\bar{h}_{1-n}e^{-in\omega}$$

$$= -e^{-i\omega}\overline{m_0(\omega+\pi)}. \tag{3-9}$$

例1　哈尔(Haar)基．设V_0由所有在$[n, n+1)$(n为整数)上取常值的分段常值函数$f(t) \in L^2(\mathbf{R})$构成．$V_j = \{f(2^j t) \mid f(t) \in V_0\}$，那么$\{V_j\}_{j=-\infty}^{+\infty}$为$L^2(\mathbf{R})$的闭子空间增加列．再取$\varphi(t)$为区间$[0,1)$的特征函数，于是不难验证$(\{V_j\}, \varphi)$构成$L^2(\mathbf{R})$的一个多尺度分析，使得$\{\varphi(t-n)\}_n$为$V_0$的标准正交基．易验证，

$$\hat{\varphi}(\omega) = \frac{1}{i\omega}(1 - e^{-i\omega}),$$

$$m_0(\omega) = \frac{\hat{\varphi}(2\omega)}{\hat{\varphi}(\omega)} = \frac{1}{2}(1 + e^{-i\omega}),$$

$$m_1(\omega) = -e^{i\omega}\overline{m_0(\omega+\pi)} = \frac{1}{2}(1 - e^{-i\omega}),$$

所以相应的基本小波为

$$\hat{\psi}(\omega) = m_1\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right) = \frac{1}{2}(1 - e^{i\omega/2})\hat{\psi}\left(\frac{\omega}{2}\right),$$

$$\psi(t) = \varphi(2t) - \varphi(2t-1);$$

相应的双尺度方程为

$$\varphi(t) = \varphi(2t) + \varphi(2t-1);$$

$\{\psi_{mn}(t)\}$构成$L^2(\mathbf{R})$的正交小波基，即哈尔基．

例2　**Littlewood-Paley基**．令$\varphi(t) = \sin\pi t/\pi t$，

$$V_0 = \{f(t) \in L^2(\mathbf{R}) \mid \operatorname{supp}\hat{f} \subseteq [-\pi, \pi]\},$$

$$V_j = \{f(t) \in L^2(\mathbf{R}) \mid f(2^{-j}t) \in V_0\}$$

$$= \{f(t) \in L^2(\mathbf{R}) \mid \mathrm{supp}\hat{f} \subseteq [-2^j\pi, 2^j\pi]\}.$$

注意到 $\hat{\varphi}(\omega) = \chi_{[-\pi,\pi]}(\omega)$. 不难验证 $\varphi(t) \in V_0$ 且 $\{\varphi(t-n)\}_{n=-\infty}^{\infty}$ 构成 $L^2(\mathbf{R})$ 中的标准正交系．再由取样定理,对任何 $f(t) \in V_0$,

$$f(t) = \sum_n f(n)\varphi(t-n).$$

从而知($\{V_j\}, \varphi$) 为 $L^2(\mathbf{R})$ 的多尺度分析．

$$m_0(\omega) = \chi_{[-\frac{\pi}{2},\frac{\pi}{2}]}(\omega) \text{ 的 } 2\pi \text{ 周期化},$$

$$\hat{\psi}(\omega) = -e^{-i\omega/2}\chi_A(\omega), \quad A = [-2\pi, -\pi] \cup [\pi, 2\pi],$$

$$\psi(t) = \frac{1}{\pi}[\sin\pi(t-\frac{1}{2}) - \sin 2\pi(t-\frac{1}{2})][t-\frac{1}{2}]^{-1}.$$

3.1.4　多尺度分析的一个等价定义

在多尺度分析的定义1中,可以将条件4°减弱为

5°　$\{\varphi(t-n)\}_n$ 构成 V_0 的里斯基,即每个 $f \in V_0$ 都可由 $\{\varphi(t-n)\}_n$ 表出:

$$f(t) = \sum_n c_n\varphi(t-n),$$

且存在与 $f(t) \in V_0$ 无关的正数 A、B 使得

$$A\sum_n |c_n|^2 \leqslant \|f\|_2^2 \leqslant B\sum_n |c_n|^2, \quad \forall f \in V_0. \tag{3-10}$$

事实上,利用下列命题:

命题3　不等式(3-10)成立当且仅当

$$A \leqslant 2\pi\sum_n |\hat{\varphi}(\omega + 2n\pi)|^2 \leqslant B. \tag{3-11}$$

取 $\varphi^*(t)$ 为

$$\widehat{\varphi^*}(\omega) = \hat{\varphi}(\omega)(2\pi\sum_n |\hat{\varphi}(\omega+2n\pi)|^2)^{-1/2}, \tag{3-12}$$

则 $\varphi^*(t) \in L^2(\mathbf{R})$. 若记

$$(2\pi\sum_n |\hat{\varphi}(\omega+2n\pi)|^2)^{-1/2} = \sum_n d_n e^{-in\omega},$$

那么

$$\widehat{\varphi^*}(\omega) = \sum_n d_n e^{-in\omega}\hat{\varphi}(\omega),$$

$$\sum_n |\widehat{\varphi^*}(\omega+2n\pi)|^2 = \frac{1}{2\pi}.$$

进而易知 $\{\varphi^*(t-n)\}_n$ 构成 V_0 的标准正交基．

(3-12)式可以看作对尺度函数 $\varphi(t)$ 进行"正交化"而得到对应于正交小波基的尺度函数 φ^*．

例3　B-样条小波．记 $N_1(t) = \chi_{[0,1]}(t)$, $N_k(t) = (N_{k-1} * N_1)(t)(k \geqslant 2)$ 为 k 阶 B 样条,则

$$\hat{N}_k(\omega) = \sqrt{2\pi}^{-1}\left(\frac{1-e^{-i\omega}}{i\omega}\right)^k = 2^{-k}(1+e^{-i\omega/2})^k\hat{N}_k(\frac{\omega}{2}).$$

所以 $N_k(t)$ 满足下列双尺度方程:

$$N_k(t) = 2^{1-m}\sum_{j=0}^{k}\binom{k}{j}N_k(2t-j).$$

虽然$\{N_k(t-n)\}_n$并不是正交系，但它却是它所张成的空间 V_0 的里斯基．事实上，若记

$$F(\omega) = \sum_n |\hat{N}_k(\omega+2n\pi)|^2 = \sum_n \eta(n)e^{-in\omega},$$

则不难算得

$$\eta(t) = \int_{-\infty}^{\infty} N_k(t+u)N_k(u)du = \eta(-t).$$

从而由 $N_k(t)$ 的紧支集性质得 $F(\omega)$ 为 m 阶三角多项式．再考察 $\hat{N}_k(\omega)$ 的零点，即知 $F(\omega)$ 无零点．根据命题3-3便知$\{N_k(t-n)\}_n$为 V_0 的里斯基．

进一步对 $N_k(t)$ 实施"正交化"，在数值上是可行的．此外，正交化后得到的正交小波具有某些对称性，具有指数衰减速度，还可具有较好的正则性(即光滑性)．参见文献[2]．

3.1.5 通过滤波函数 $m_0(\omega)$ 构造多尺度分析与正交小波基

命题4 设 $m_0(\omega) = \sum_{k=-\infty}^{+\infty} c_k e^{-ik\omega}$，满足

1° $|c_k| = O((1+|k|^2)^{-1})$；

2° $|m_0(0)| = 1$；

3° $|m_0(\omega)|^2 + |m_0(\omega+\pi)|^2 = 1$；

4° $m_0(\omega) \neq 0, \omega \in [-\pi/2, \pi/2]$．

若定义 $\varphi(t)$ 为

$$\hat{\varphi}(\omega) = \prod_{k=1}^{\infty} m_0(2^{-k}\omega);$$

那么$\{\varphi(t-n)\}_{n=-\infty}^{\infty}$为 $L^2(\mathbf{R})$ 中标准正交系．定义

$$V_0 = \overline{\operatorname{span}\{\varphi(t-n)\}_n},$$

$$V_j = \{f(2^jt) \mid f(t) \in V_0\}, \quad j = 0, \pm 1, \pm 2, \cdots,$$

则$(\{V_j\}, \varphi)$构成 $L^2(\mathbf{R})$ 的一个多尺度分析．

命题5 设 $m_0(\omega)$ 为 2π- 周期的连续函数，满足

$$|m_0(\omega)|^2 + |m_0(\omega+\pi)|^2 = 1, \quad m_0(0) = 1,$$

且无穷乘积 $\prod_1^{\infty} m(2^{-k}\omega)$ 几乎处处收敛于连续函数 $\hat{\varphi}(\xi)$．那么 $\hat{\varphi}(\omega) \in L^2(\mathbf{R})$．此外，$\{\varphi(t-k)\}$为 $L^2(\mathbf{R})$ 中标准正交系的充要条件是存在常数 $c>0$，使得

$$\sum_k |\hat{\varphi}(\omega+2k\pi)|^2 \geq c \tag{3-13}$$

几乎处处成立．

命题6 在命题5中，若再假定 $G(\omega) = \sum_k |\hat{\varphi}(\omega+2k\pi)|^2$ 连续，则条件

(3-13) 可以换为科恩(Cohen)条件:存在与单位圆周 T 同余的紧集 K,以 O 为内点,使得

$$\inf\{|m_0(2^{-j}\omega)| \,\big|\, j \geqslant 1, \omega \in K\} \geqslant c_0 > 0.$$

其中称 K 与 T 同余是指 $|K| = 2\pi$,且对任何 $\omega \in T$,存在整数 j 使 $\omega + 2j\pi \in K$.

更详尽的讨论见文献[3].

3.1.6 没有对应的多尺度分析的正交小波基

虽然目前人们掌握的正交小波基绝大部分都出自于多尺度分析. 但下面的正交小波函数 $\psi(t)$ 却不能产生于某个多尺度分析. 事实上它无对应的尺度函数 $\varphi(t)$,

$$\hat{\psi}(\omega) = \begin{cases} \frac{1}{2\pi}, & 4\pi/7 \leqslant |\omega| \leqslant \pi \text{ 或 } 4\pi \leqslant |\omega| \leqslant 32\pi/7, \\ 0, & \text{其它}. \end{cases}$$

可验证

$$\sum_k |\hat{\psi}(\omega + 2k\pi)|^2 = \frac{1}{2\pi},$$

但却不存在相应于 $\psi(t)$ 的尺度函数 $\varphi(t)$.

3.2 信号的小波分解与合成,Mallat 算法

3.2.1 信号的小波分解

设$(\{V_j\}, \varphi)$ 为 $L^2(\mathbf{R})$ 的一个多尺度分析,$\psi(t)$ 为对应的正交小波.

1. 信号的小波分解

给定信号 $f(t) \in L^2(\mathbf{R})$,由于实际取样的原因,总可假定 $f(t)$ 是频限信号,即存在 N 使

$$f \in V_N = \bigoplus_{k=-\infty}^{N-1} W_k.$$

不妨假定 $f(t) \in V_0$,则有

$$f(t) = \sum_n c_n^0 \varphi_{0n}(t).$$

令 P_j 为 V_{-j} 上的投影算子,Q_j 为 W_{-j} 上的投影算子,则有

$$f_j = \mathrm{P}_j f = \sum_n c_n^j \varphi_{-jn} \in V_{-j}, \quad j \geqslant 0,$$

$$\mathrm{Q}_j f = \sum_n d_n^j \psi_{-jn} \in W_{-j}, \quad j \geqslant 1.$$

2. 小波分解系数的计算

利用(3-2)式及(3-7)式可以算得

$$c_n^j = \langle f_j, \varphi_{-jn} \rangle = \langle f_{j-1}, \varphi_{-jn} \rangle$$

$$= \sum_k \bar{h}_{k-2n} c_k^{j-1},$$

$$d_n^j = \langle Q_j f, \psi_{-jn} \rangle = \langle f_{j-1}, \psi_{-jn} \rangle = \sum_k \bar{g}_{k-2n} c_k^{j-1}.$$

3. 分解算子 H、G 与分解图式

记 $C^j = \{C_n^j\}_n, D^j = \{d_n^j\}_n$,又记 H、G 分别为如下定义的 l^2 上的线性算子:

$$\begin{aligned} (\mathrm{H}a)_n &= \sum_k \bar{h}_{k-2n} a_k, \quad a = (a_k) \in l^2, \\ (\mathrm{G}a)_n &= \sum_k \bar{g}_{k-2n} a_k, \quad a = (a_k) \in l^2, \end{aligned} \tag{3-14}$$

则上面计算可表述为

$$C^j = \mathrm{H} C^{j-1}, \quad D^j = \mathrm{G} C^{j-1}, \quad j \geq 1.$$

上面分解模式可描述为

$$\begin{array}{ccccccccccc} C^0 & \xrightarrow{H} & C^1 & \xrightarrow{H} & C^2 & \xrightarrow{H} & \cdots & \xrightarrow{H} & C^{N-1} & \xrightarrow{H} & C^N \\ & \searrow^{G} & & \searrow^{G} & & & & & & \searrow^{G} & \\ & & D^1 & & D^2 & & \cdots & & & & D^N \end{array}$$

它的实际意义是:D^1 为 C^0 的高频部分,C^1 为 C^0 的低频部分;第二步再将 C^1 分解为高频部分 D^2 与低频部分 C^2,最后分解到第 N 步 D^N 与 C^N.

4. 采样初始化问题

上述算法需要首先确定信号 $f(t)$ 的小波分解系数列 $C^0 = \{c_n^0\}$. 若给定的是 $f(t)$ 的离散采样 $\{f(n)\}_n$,则一个简单的方法就是将 $\{f(n)\}_n$ 等同于 C^0. 这在许多情况下是可行的,否则就需要采用其它采样初始化算法求得 $C^0 = \{c_n^0\}$.

3.2.2 信号的合成

1. 合成算子 H*、G* 与合成图式

令 H*、G* 分别为 H 与 G 的共轭算子:

$$\left.\begin{aligned} (\mathrm{H}^* a)_n &= \sum_k \bar{h}_{n-2k} a_k, \\ (\mathrm{G}^* a)_n &= \sum_k \bar{g}_{n-2k} a_k, \end{aligned}\right\} \quad a = (a_k) \in l^2. \tag{3-15}$$

那么

$$C^{j-1} = \mathrm{H}^* C^j + \mathrm{G}^* D^j, \quad j = 1, 2, \cdots, N,$$

所以,

$$\begin{aligned} C^0 = \mathrm{G}^* D^1 + \mathrm{H}^* \mathrm{G}^* D^2 + (\mathrm{H}^*)^2 \mathrm{G}^* D^3 + \cdots + \\ (\mathrm{H}^*)^{N-1} \mathrm{G}^* D^N + (\mathrm{H}^*)^N C^N. \end{aligned}$$

上述信号的小波分解与合成算法常称为 **Mallat 算法**.

2. 关于 Mallat 算法的数据量

若原始数据 $C^0 = \{c_n^0\}$ 的数据量为 M,在不计边界影响的情况下,$D^1, D^2, \cdots, D^N$ 及 C^N 中所含的数据总量仍为 M.

4　紧支集正交小波基

4.1　双尺度方程的具紧支集解

4.1.1　滤波函数的构造

设($\{V_j\}$, φ)为 $L^2(\mathbf{R})$ 的一个多尺度分析，$\varphi(t)$ 满足双尺度方程

$$\varphi(t) = \sum_n c_n \varphi(2t - n),$$

$\varphi(t)$ 对应的滤波函数 $m_0(\omega)$ 为

$$m_0(\omega) = \frac{1}{2}\sum_n c_n e^{-in\omega}.$$

1. 紧支集尺度函数的特性

若尺度函数 $\varphi(t)$ 具有紧支集，例如，$\mathrm{supp}\varphi \subseteq [0, a]$，则 $\mathrm{supp}\varphi(2x - n) \subseteq [\frac{n}{2}, \frac{n}{2} + \frac{a}{2}]$. 从而当 $n \geqslant 2a$ 或 $n < -a$ 时，

$$c_n = \int_{-\infty}^{\infty} \varphi(t)\overline{\varphi}(2t - n)\mathrm{d}t = 0.$$

所以小波函数 $\psi(t)$ 亦有紧支集并且滤波函数 $m_0(\omega)$ 为三角多项式．从3.2节的讨论就可以看出，这对于信号的小波分解与合成是很有好处的，且不为零的系数 c_n 的个数越少计算越简单．

2. 对应于紧支集尺度函数的滤波函数 $m_0(\omega)$

由(3-5)式知，$m_0(\omega)$ 应满足

$$|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = 1. \tag{4-1}$$

此外，为使 $\varphi(t)$ 与 $\psi(t)$ 有足够的正则性(即光滑性)，以下设 $m_0(\omega)$ 为具下列形式的三角多项式：

$$m_0(\omega) = [2^{-1}(1 + e^{i\omega})]^N \mathscr{L}(\omega), \tag{4-2}$$

其中 $N \geqslant 1$, $\mathscr{L}(\omega)$ 为三角多项式且 $\mathscr{L}(0) = 1$, $\mathscr{L}(\pi) \neq 0$.

为寻找具有紧支集的尺度函数 $\varphi(t)$，首先寻找具有形状(4-2)且满足(4-1)式的 $m_0(\omega)$.

命题1　具有形状(4-2)且满足(4-1)式的 $m_0(\omega)$ 可写为

$$|m_0(\omega)|^2 = \left(\cos^2\frac{\omega}{2}\right)^N\left[\sum_{k=0}^{N-1}\binom{N-1+k}{k}\left(\sin\frac{\omega}{2}\right)^{2k} + \left(\sin\frac{\omega}{2}\right)^{2N} r\left(\sin\frac{2\omega}{2}\right)\right].$$

其中 $r(x)$ 为任一代数多项式，满足

$$r(x) + r(1 - x) = 0.$$

进而只需选择适当的多项式 $r(x)$，使上式右端为正，再将上式右端"开方"求出 $m_0(\omega)$ 即可．由于上式右端为偶三角多项式，因而这是可行的，并且可以找到 m_0 的构造方法，详见文献[2]．

3．对应于 $m_0(\omega)$ 的 $\varphi(t)$ 的构造

命题 2　设 $m_0(\omega)$ 具形状(4-2)且满足(4-1)式，则无穷乘积 $\prod_1^{\infty} m_0(2^{-j}\omega)$ 在每个紧集上一致收敛于一个具有紧支集的连续函数 $\hat{\varphi}(\omega)$，并且

$$\|\hat{\varphi}\|_2 \leqslant 1, \quad |\hat{\varphi}(\omega)| \leqslant 1, \quad \omega \in \mathbf{R}.$$

此时显然有 $\hat{\varphi}(\omega) = m_0(\omega/2)\hat{\varphi}(\omega/2)$．若设 m_0 为

$$m_0(\omega) = \frac{1}{2}\sum_0^k c_n e^{-in\omega}, \tag{4-3}$$

则 $\varphi(t)$ 满足有限长的双尺度方程

$$\varphi(t) = \sum_{n=0}^{k} c_n \varphi(2t - n). \tag{4-4}$$

需要指出，命题 4-2 中给出的 $\varphi(t)$ 还不一定能成为某个多尺度分析的尺度函数，在 4.2 节中将进一步讨论．

4.1.2　迭代法求解有限长双尺度方程

(1) 为求解有限长双尺度方程(4-4)，任选 $\varphi_0(t) \in L^2(\mathbf{R})$，使得 $\hat{\varphi}(\omega)$ 连续且 $\hat{\varphi}(0) = 1$．令

$$\varphi_n(t) = \sum_{n=0}^{k} c_n \varphi_{n-1}(2t - n), k = 1, 2, \cdots,$$

则有

$$\hat{\varphi}_n(\omega) = m_0\left(\frac{\omega}{2}\right)\hat{\varphi}_{n-1}\left(\frac{\omega}{2}\right) = \prod_{j=1}^{k} m_0(2^{-j}\omega)\hat{\varphi}_0(2^{-k}\omega),$$

其中 $m_0(\omega)$ 如(4-3)式．若方程(4-4)满足

$$\sum_{n=0}^{k} c_n = 2,$$

即 $m(0) = 1$，则可以证明，无穷乘积 $\prod_{j=1}^{+\infty} m_0(2^{-j}\omega)$ 点态收敛于连续函数 $\hat{\varphi}(\omega)$．通常需要再对 $m_0(\omega)$ 附加某些条件，才能保证 $\prod_{j=1}^{\infty} m_0(2^{-j}\omega)$ 在 $L^2(\mathbf{R})$ 中亦收敛．这里假定

$$\prod_{j=1}^{\infty} m_0(2^{-j}\omega) = \hat{\varphi}(\omega) \in L^2(\mathbf{R}).$$

那么

$$\lim_{n\to\infty}\widehat{\varphi}_n(\omega) = \widehat{\varphi}(\omega) \in L^2(\mathbf{R}).$$

显然，$\varphi(t)$ 为方程(4-4) 的解．

(2) 在(1) 中，$\varphi_0(t)$ 的选取自由度相当大．最后都导致同一个 $\varphi(t)$．若选取 $\varphi_0(t) = N_2(t) = \chi_{[0,1]} * \chi_{[0,1]}(t)$ 为二阶 B- 样条，则得

$$\mathrm{supp}\varphi_0 = [0,2], \quad \mathrm{supp}\varphi_1 = [0,\frac{2+k}{2}], \cdots,$$

$$\mathrm{supp}\varphi_n = [0,\frac{2+k(2^n-1)}{2^n}], \cdots$$

令 $n\to+\infty$，即得 $\mathrm{supp}\varphi = [0,k]$．

4.2 $\varphi(t)$ 成为尺度函数的条件

4.2.1 $\varphi(t)$ 导出紧支集小波框架

4.1 中构造出的具有紧支集的 $\varphi(t)$ 虽然满足双尺度方程(4-4)，但却不一定为某个多尺度分析的尺度函数，然而仍有：

定理1 设三角多项式 $m_0(\omega)$ 满足(4-1) 与(4-2) 式，$\widehat{\varphi}(\omega) = \prod_1^\infty m_0(2^{-j}\omega)$，$\psi(t)$ 由(3-7) 式给出．那么 $\{\psi_{mn}\}_{mn}$ 构成 $L^2(\mathbf{R})$ 的一个紧小波框架且以 1 为其框架常数，即

$$\sum_{mn} |\langle f,\psi_{mn}\rangle|^2 = \|f\|_2^2, \quad f\in L^2(\mathbf{R}).$$

其证明见文献[2]．

要使 4.1 中构造的 $\varphi(t)$ 成为尺度函数，当且仅当

$$\sum_n |\widehat{\varphi}(\omega+2n\pi)|^2 \equiv 1. \tag{4-5}$$

下面的科恩(Cohen) 条件也是 $\varphi(t)$ 成为尺度函数的充要条件：

存在与$[-\pi,\pi]$ 同余的紧集 K，以 O 为其内点且满足

$$\inf_{j>0}\inf_{\omega\in K} |m_0(2^{-j}\omega)| > 0.$$

[2,6.2 和 6.3] 中给出了另一些条件．

例1 (I. Danbechies) 的紧支集小波$_N\varphi$) 在命题4-1 中取多项式 $r(x) = 0$，即取

$$|{}_Nm_0(\omega)|^2 = (\cos^2\frac{\omega}{2})^N \sum_{k=0}^{n-1}\binom{N-1+k}{k}(\sin\frac{\omega}{2})^{2k},$$

由此得到一组紧支集正交小波$_N\varphi$、$_N\psi$．[2,6.4] 给出了这组正交小波的性质以及双尺度方程

$$_N\psi(t) = \sqrt{2}\sum_{n=0}^{2N-1} {}_Nh_n\, {}_N\varphi(2t-n)$$

中的系数$\{_N h_n\}$($N = 2,3,\cdots,10$)的值.

5 小波包

5.1 小波包的构造

5.1.1 小波包的概念

设$(\{V_j\},\varphi)$为$L^2(\mathbf{R})$上的多尺度分析,$\{\varphi(t-n)\}_n$为V_0的标准正交基,

$$\varphi(x) = \sqrt{2}\sum_n h_n\varphi(2x-n).$$

为简单计,以下设$\{h_n\}$为实数列.从而相应的正交小波为

$$\psi(t) = \sqrt{2}\sum_n g_n\varphi(2x-n), \quad g_n = (-1)^n h_{1-n}.$$

令 $W_0(t) = \varphi(t)$,$W_1(t) = \psi(t)$.而当$n \geqslant 1$时,

$$\begin{cases} W_{2n}(t) = \sqrt{2}\sum_k h_k W_n(2t-k), \\ W_{2n+1}(t) = \sqrt{2}\sum_k g_k W_n(2t-k). \end{cases} \tag{5-1}$$

定义 1 称(5-1)式确定的$\{W_n(t)\}_{n=0}^{\infty}$为由尺度函数$\varphi$确定的小波包.

例 1 $\varphi(t) = \chi_{[0,1]}(t)$为哈尔基对应的尺度函数.$\varphi(x) = \varphi(2x) + \varphi(2x-1)$,$\psi(x) = \varphi(2x) - \varphi(2x-1)$,即

$$h_0 = h_1 = g_0 = -g_1 = \frac{1}{\sqrt{2}}.$$

所以,

$$\begin{aligned} W_2(t) &= W_1(2t) + W_1(2t-1) \\ &= \varphi(4t) - \varphi(4t-1) + \varphi(4t-2) + \varphi(4t-3), \\ W_3(t) &= W_1(2t) - W_1(2t-1) \\ &= \varphi(4t) - \varphi(4t-1) - \varphi(4t-2) + \varphi(4t-3), \\ W_4(t) &= W_2(2t) + W_2(2t-1) = \cdots. \end{aligned}$$

由例1可以给$W_n(t)$一个直观的描述:小波包$\{W_n(t)\}_n$中每个函数$W_n(t)$有大致相同的"尺度",但$W_n(t)$振荡的次数与n成正比,即n越大,$W_n(t)$的频率越高.

5.1.2 小波包的性质

令Ω_n为$\{W_n(t-k)\}_{k=-\infty}^{\infty}$张成的闭子空间,则有

1° $\{W_n(t-k)\}_{k=-\infty}^{\infty}$($n \geqslant 0$)为$\Omega_n$的标准正交基.

2° 在$L^2(\mathbf{R})$上定义伸缩算子δ:

$$\delta f(t) = \sqrt{2} f(2t), \quad f \in L^2(\mathbf{R}).$$

则 $\Omega_{2n} \perp \Omega_{2n+1}$ 并且 $\delta\Omega_n = \Omega_{2n} \oplus \Omega_{2n+1}$.

3° 由于 $V_0 = \Omega_0$,根据多尺度分析定义,

$$\delta^k\Omega_0 = \delta^k V_0 = \delta^{k-1}(\delta V_0) = \delta^{k-1} V_1 = V_k,$$

所以 V_k 有下列直交分解:

$$V_k = \delta^k\Omega_0 = \delta^{k-1}(\Omega_0 \oplus \Omega_1) = \cdots$$
$$= \Omega_0 \oplus \Omega_1 \oplus \cdots \oplus \Omega_{2^k-1}.$$

4° 由1°、3°便知$\{W_n(t-j) \mid n \geqslant 0, j$为整数$\}$构成了$L^2(\mathbf{R})$的标准正交基.进而得,对每个整数$k \geqslant 0$,$\{\sqrt{2}^k W_n(2^k t - j) \mid n \geqslant 0, j$为整数$\}$都构成$L^2(\mathbf{R})$的标准正交基.

5.1.3　小波库与小波包基

定义2　由尺度函数$\varphi(t)$导出的函数族$\{\sqrt{2}^k W_n(2^k t - j) \mid n \geqslant 0, k \geqslant 0, j$为整数$\}$称为$\varphi(t)$导出的小波库.在此小波库中选出的任一个$L^2(\mathbf{R})$的标准正交基都称为$L^2(\mathbf{R})$的小波包基.

5.1.4　小波包基的表示

将非负整数集$\mathbf{N}^+$分解为子集

$$I_{nk} = \{2^k n, 2^k n + 1, \cdots, 2^k(n+1) - 1\}, \quad n, k \geqslant 0.$$

若用P表示$\mathbf{N}^+$,用这样的I_{nk}所作的一个分解,即:$\mathbf{N}^+ = \bigcup \{I_{nk} \mid I_{nk} \in P\}$且$P$中$I_{nk}$互不相交,那么$\{\sqrt{2}^k W_n(2^k t - j) \mid I_{nk} \in P, j \in \mathbf{Z}\}$为$L^2(\mathbf{R})$的一个小波包基.

由类似的推导,上面结论对$k < 0$也成立.

5.2　信号的小波包基展开

5.2.1　信号在小波包基下的系数计算

设多尺度分析$(\{V_j\}, \varphi)$及相应的小波库$\{\sqrt{2}^k W_n(2^k t - j) \mid n \geqslant 0, k \geqslant 0, j \in \mathbf{Z}\}$如5.1.设$x(t)$为$L^2(\mathbf{R})$中的一个频限信号,不妨设$x(t) \in V_L$.对于$0 \leqslant k \leqslant L$,$0 \leqslant n < 2^{L-k}$,$\delta^k\Omega_n \subseteq V_L$.令$x(t)$在$\delta^k\Omega_n$上的正交投影为$x^{nk}(t)$,则有

$$x^{nk}(t) = \sqrt{2}^k \sum_j c_j^{nk} W_n(2^k t - j).$$

若记$C^{nk} = \{c_j^{nk}\}_j$,H与G为(3-14)式中定义的l^2上的线性算子,那么可以算得

$$HC^{nk} = C^{2n,k-1}, GC^{nk} = C^{2n+1,k-1},$$
$$0 \leqslant n < 2^{L-k+1}, 1 \leqslant k \leqslant L.$$

5.2.2　离散信号的分解图式

设已知$C^{0L} = \{c_j^{0L}\}_j$,则可利用H、G得到$C^{0,L-1}$及$C^{1,L-1}$,第二步再将H、G作

用于 $C^{0,L-1}$ 及 $C^{1,L-1}$,依此方法便得到如下分解图式图 5-1.

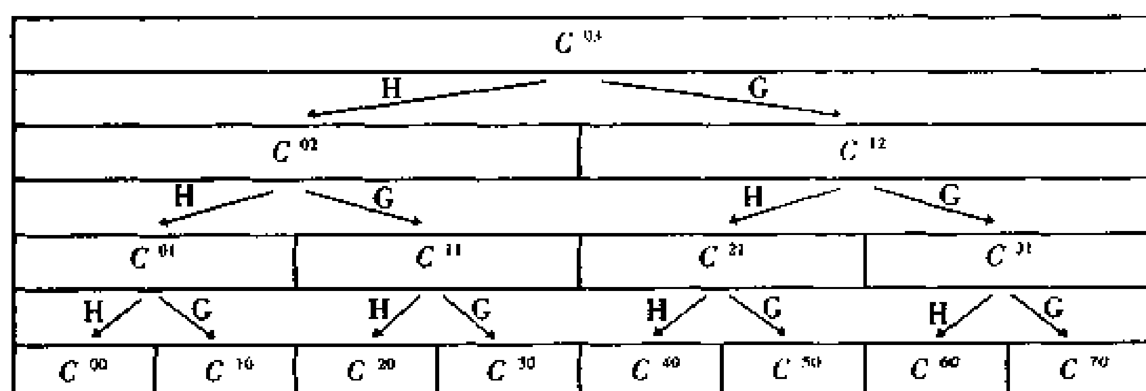


图 5-1 （$L=3$）

在上面分解模式中,若 $C^{nk}=\{C_j^{nk}\}_{j=1}$ 只有 $2M$ 项不为零,则通常将 C^{nk} 作周期延拓．于是对于 $C^{2n,k-1}$ 与 $C^{2n+1,k-1}$,可以算得

$$C_{j+M}^{2n,k-1}=C_j^{2n,k-1},\ C_{j+M}^{2n+1,k-1}=C_j^{2n+1,k-1}.$$

即此时 $C^{2n,k-1}$ 与 $C^{2n+1,k-1}$ 本质上只有 M 项不为零．

5.2.3 信号的合成

若已知 $C^{2n,k-1}$ 与 $C^{2n+1,k-1}$,则利用 H 与 G 的共轭算子 H^* 与 G^*（见(3-15)式）即可恢复出 C^{nk}:

$$C^{nk}=H^*C^{2n,k-1}+G^*C^{2n+1,k-1}.$$

5.3 最优基的选择

5.3.1 信息花费函数

为了确定选择最优基的标准,我们引入信息花费函数的概念．

称从 l^2 到 $\mathbf{R}$ 的映射 M 为一个**信息花费函数**,若 $M(0)=0$ 并且有某个 $\mathbf{R}$ 上的函数 $m(x)$ 使得对 l^2 中任意数列 $\{x_j\}$ 有

$$M(\{x_j\})=\sum_j m(x_j)\quad（可加性）.$$

选择信息花费函数的一般准则是:当 $\{x_j\}$ 的"质量"集中时,即集中在较少分量上时,$M(\{x_j\})$ 应较小,而当 $\{x_j\}$ 的分量大小较平均时,$M(\{x_j\})$ 应较大．下面是一些常用的信息花费函数．

例 2 $M(\{x_j\})$ 为 $\{x_j\}$ 中满足 $|x_j|>\varepsilon$ 的分量个数,其中 $\varepsilon>0$ 预先确定．

例 3 $M(\{x_j\})=\sum_j |x_j|^p, 0<p\leqslant 2.$

例 4 （对数熵）$M(\{x_j\})=\sum_j \log|x_j|^2$,其中约定 $\log 0=0$.

5.3.2　最优基的概念

设给定信号 $f(t) \in L^2(\mathbf{R})$，M 为一个信息花费函数．在小波库中任取小波包基 B，记 Bf 为 f 在基 B 下的分解系数列，则可定义 f 关于 M 的最优基为使得 $M(Bf)$ 最小的小波包基 B.

5.3.3　最优基的选取

令 $B_{nk} = \{\sqrt{2}^k W_n(2^k t - j)\}_j$，它是 $\delta^k \Omega_n$ 的标准正交基．又记 A_{nk} 为 $f(t)$ 在 $\delta^k \Omega_n$ 中的正交投影关于 M 的最优基．则

当 $k = 0$ 时，Ω_n 只有一个基 B_{n0}．故 $A_{n0} = B_{n0}$.

当 $k = 1$ 时，取

$$A_{n1} = \begin{cases} B_{n1}, & 若\ M(B_{n1}f) < M(A_{2n,0}f) + M(A_{2n+1,0}f), \\ A_{2n,0} \cup A_{2n+1,0}, & 否则． \end{cases}$$

设 A_{nk}，$n = 0,1,2,\cdots$，都已选定．则选取

$$A_{n,k+1} = \begin{cases} B_{n,k+1}, & 若\ M(B_{n,k+1}f) < M(A_{2n,k}f) + M(A_{2n+1,k}f), \\ A_{2nk} \cup A_{2n+1,k}, & 否则． \end{cases}$$

不难验证，上面算法将选出 $f(t)$ 关于 M 的最优基．

参 考 文 献

1　Chui C K. An introduction to wavelets. New York: Academic Press, 1992.

2　Daubechies I. Ten lectures on wavelets, CBMS-Conference Lecture Notes. Vol. 61. Philadelphia: SIAM, 1992.

3　龙瑞麟．高维小波分析．北京：世界图书出版公司，1995.

第 8 篇

Petri 网

编　者　袁崇义
审校者　陆维明

目　录

引　言

Petri 网是没有任何全局控制的模型，适合于描述和分析异步并发系统.

1962 年 Petri(Carl Adam Petri)在他的博士论文中首先使用了后来以他的名字命名的系统模型.这种模型由变迁和状态两类元素构成.可以用网状结构图示，因而称为 Petri 网.20 世纪 60—70 年代的 Petri 网研究以开发网系统分析技术为中心；70 年代末开始的理论研究产生了以并发论、同步论、网逻辑和网拓扑为主要内容的通用网论.

从基本网系统、P/T-系统、高级网系统到自控系统，Petri 网构成完整的层次结构，适合于从理论研究、分析技术开发到实际应用的不同要求.20 世纪 80 年代中期以来，Petri 网已广泛应用于网络通信、软件工程、系统性能分析、数据库管理，MIS(系统)及 FMS 等.随着并行技术的发展，Petri 网得到越来越多领域中人们的重视，已成为并行处理研究的重要工具之一.

1 有　向　网

Petri 网着眼于以资源(物质资源和信息资源)流动为特征的系统.用有向弧把状态元素和变迁元素联系起来，表示资源流动的方向，就得到系统的网结构.

1.1　有向网结构

1.1.1　有向网概念

三元组 $N=(S,T;F)$ 称为**有向网**的条件是：

$$S\cap T=\varnothing\wedge S\cup T\neq\varnothing\wedge F\subseteq S\times T\cup T\times S\wedge \mathrm{dom}(F)\cup\mathrm{cod}(F)=S\cup T,$$

其中

$$\mathrm{dom}(F)=\{x\mid\exists y\in S\cup T,(x,y)\in F\},$$
$$\mathrm{cod}(F)=\{y\mid\exists x\in S\cup T,(x,y)\in F\}.$$

S、T 分别是 N 的状态元素集和变迁元素集，F 是流关系，$\mathrm{dom}(F)$ 和 $\mathrm{cod}(F)$ 分别为 F 的定义域和值域.通常记 $X=S\cup T$，称为 N 的元素集.

1.1.2　对偶网和逆网

$N'=(T,S;F)$ 称为有向网 $N=(S,T;F)$ 的**对偶网**.

$N'=(S,T;F^{-1})$ 称为有向网 $N=(S,T;F)$ 的**逆网**，其中 $F^{-1}=\{(x,y)\mid(y,$

$x) \in F\}$.

1.1.3 简单网、纯网和子网

令 $x \in X$ 为有向网 $N = (S,T;F)$ 之任一元素，

$$^{\bullet}x = \{y \mid (y,x) \in F\} \text{ 称为 } x \text{ 的前集},$$

$$x^{\bullet} = \{y \mid (x,y) \in F\} \text{ 称为 } x \text{ 的后集}.$$

有向网 $N = (S,T;F)$ 称为**简单网**的条件是：$\forall x、y \in X$,

$$^{\bullet}x = {^{\bullet}y} \wedge x^{\bullet} = y^{\bullet} \Rightarrow x = y.$$

有向网 $N = (S,T;F)$ 称为**纯网**的条件是：$\forall x \in X, {^{\bullet}x} \cap x^{\bullet} = \varnothing$.

$N' = (S',T';F')$ 称为有向网 $N = (S,T;F)$ 的**子网**，如果

$$S' \subseteq S \wedge T' \subseteq T \wedge F' = F \cap (S' \times T' \cup T' \times S').$$

1.1.4 S_-图，S_-网，T_-图和T_-网

有向网 $N = (S,T;F)$ 若满足 $\forall t \in T, |{^{\bullet}t}| = |t^{\bullet}| = 1$，则称为 **$S$图**，其中 $|{^{\bullet}t}|$ 和 $|t^{\bullet}|$ 分别是 t 的前集 $^{\bullet}t$ 和后集 $t^{\bullet}$ 的元素个数；若 $\forall t \in T, |{^{\bullet}t}| \leqslant 1 \wedge |t^{\bullet}| \leqslant 1$，则称为 **$S$网**.

若有向网 $N = (S,T;F)$ 满足 $\forall s \in S, |{^{\bullet}s}| = |S^{\bullet}| = 1$，则称为 **$T$图**；若满足 $|{^{\bullet}S}| \leqslant 1 \wedge |s^{\bullet}| \leqslant 1$，则称为 **$T$网**.

1.1.5 自由选择网

$N = (S,T;F)$ 为自由选择网的条件是：$\forall s \in S, |s^{\bullet}| > 1 \Rightarrow {^{\bullet}(s^{\bullet})} = \{s\}$.

1.1.6 扩充的自由选择网

$N = (S,T;F)$ 为**扩充的自由选择网**的条件是：$\forall s_1、s_2 \in S, s_1^{\bullet} \cap s_2^{\bullet} = \varnothing \vee s_1^{\bullet} = s_2^{\bullet}$.

1.1.7 不对称选择网

$N = (S,T;F)$ 为**不对称选择网**的条件是：$\forall s_1、s_2 \in S, s_1^{\bullet} \cap s_2^{\bullet} = \varnothing \vee s_1^{\bullet} \subseteq s_2^{\bullet} \vee s_2^{\bullet} \subseteq s_1^{\bullet}$.

1.1.8 稚网

三元组 $N' = (S,T;F)$ 为**稚网**(Pre-Net)的条件是：

$$S \cap T = \varnothing \wedge S \cup T \neq \varnothing \wedge F \subseteq S \times T \cup T \times S.$$

1.1.9 容量函数和权函数

(1) $K: S \to N^{+} \cup \{\omega\}$ 称为有向网$(S,T;F)$的**容量函数**，其中 ω 表示无穷，

$\omega = \omega + 1 = \omega - 1$，$\mathbf{N}^+$ 为不含 0 的自然数集.

(2) $W: F \to \mathbf{N}^+$ 称为有向网 $(S,T;F)$ 上的**权函数**.

(3) $W: S \times T \cup T \times S \to \mathbf{N}$ 称为有向网 $(S,T;F)$ 的**广义权函数**的条件是:

$$\forall (x,y) \in S \times T \cup T \times S, \quad W(x,y) = 0 \Leftrightarrow (x,y) \notin F$$

其中 $\mathbf{N}$ 为包括 0 在内的自然数集.

(4) $n \times m$ 阶矩阵 $C = [c_{ij}]$ 称为有向网 $(S,T;F)$ 上的**关联矩阵**的条件是:

S 和 T 分别含 n 和 m 个元素，即 $S = \{s_1, s_2, \cdots, s_n\}$，$T = \{t_1, t_2, \cdots, t_m\}$；矩阵元素 c_{ij}，$1 \leqslant i \leqslant n$，$1 \leqslant j \leqslant m$，由下式给出

$$c_{ij} = W(t_j, s_i) - W(s_i, t_j),$$

其中 W 是 $(S,T;F)$ 上的广义权函数.

(5) $n \times m$ 阶矩阵 $I_+ = [W(t_j, s_i)]$ 和 $I_- = [W(s_i, t_j)]$ 分别称为 $(S,T;F)$ 的**正矩阵**和**负矩阵**，其中 $1 \leqslant i \leqslant n$，$1 \leqslant j \leqslant m$. 它们和关联矩阵的关系是

$$C = I_+ - I_-.$$

1.1.10 标识和图形表示

$M: S \to \mathbf{N}$ 称为有向网 $(S,T;F)$ 上的**标识**的条件是 $\forall s \in S, M(s) \leqslant K(s)$，其中 K 为容量函数.

将有向网 $(S,T;F)$ 的状态元素用圆或椭圆表示，变迁元素用方框或短粗杠表示，流关系 F 用有向弧表示，权函数和容量函数分别写在相应有向弧上和圆圈附近，标识则用相应圆圈内黑点的个数表示，这种黑点通常称为**托肯**(token)，这样有向网就可用图形表示.

例 1 令 $S = \{s_1, s_2\}$，$T = \{t_1, t_2\}$，$F = \{(s_1,t_1),(t_1,s_2),(s_2,t_2),(t_2,s_1)\}$，则 $(S,T;F)$ 满足有向网定义，它的图形表示为图 1-1.

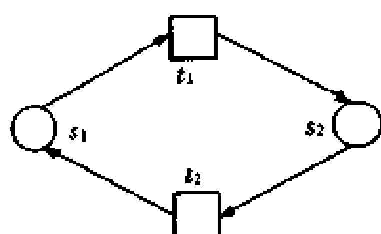


图 1-1

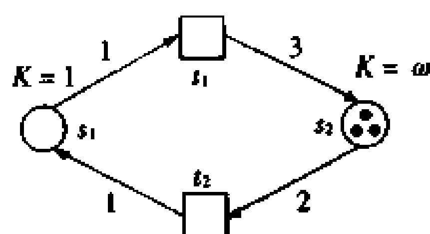


图 1-2

例 2 令 $K(s_1) = 1$，$K(s_2) = \omega$，$W(s_1,t_1) = 1$，$W(t_1,s_2) = 3$，$W(s_2,t_2) = 2$，$W(t_2,s_1) = 1$，$M(s_1) = 0$，$M(s_2) = 3$，则 K、W、M 依次为例 1 中有向网上的容量函数、权函数和标识，它们的图形表示为图 1-2.

习惯上在图示中不标出值为 1 的权和值为 ω 的容量. 图 1-2 中 $W(s_1,t_1)$ 和 $W(t_2,s_1)$ 可不写，$K(s_2) = \omega$ 也可以不写，由缺省蕴涵.

1.2 网 操 作

1.2.1 S补和T补

令$s \in S$为有向网$(S,T;F)$的状态元素，$s' \notin S \cup T$，则可以在S中增加一个以s'命名的状态元素，使$^{\cdot}s' = s^{\cdot}$，$(s')^{\cdot} = {}^{\cdot}s$，得到有向网$(S',T;F')$，其中$S' = S \cup \{s'\}$，$F' = F \cup \{(t,s') \mid (s,t) \in F\} \cup \{(s',t) \mid (t,s) \in F\}$。从$(S,T;F)$构造$(S',T;F')$的操作称为对$(S,T;F)$的$s$进行$S$补，$s'$称为$s$的补元素。

令$t \in T$为$(S,T;F)$的变迁元素，$t' \in S \cup T$，记$T' = T \cup \{t'\}$，$F' = F \cup \{(t',s) \mid (s,t) \in F\} \cup \{(s,t') \mid (t,s) \in F\}$，则有向网$(S,T';F')$为在$(S,T;F)$上对$t$作$T$补操作的结果，$t'$称为$t$的逆元素。

1.2.2 S完备化与T完备化

对有向网$(S,T;F)$做S-完备化操作，是将可以用T中元素构造的所有状态形元素添加到$(S,T;F)$中。$(S',T;F')$称为$(S,T;F)$的S-完备网，如果对任何T_1、$T_2 \subseteq T$，只要$T_1 \cup T_2 \neq \varnothing$，就有唯一的$s \in S'$，使$^{\cdot}s = T_1$，$s^{\cdot} = T_2$。具有这种性质的有向网称为**S完备的**。S完备的有向网恰有$4^m - 1$个状态元素，其中单纯状态元为$3^m - 1$个，$m = |T|$为T中元素的个数。

$(S,T';F')$称为有向网$(S,T;F)$的T_完备网，如果$\forall S_1,S_2 \subseteq S, S_1 \cup S_2 \neq \varnothing \Rightarrow \exists ! t \in T', {}^{\cdot}t = S_1 \wedge t^{\cdot} = S_2$。$(S,T';F')$称为**T完备的**，它共有$4^n - 1$变迁形元素，其中$3^n - 1$个是单纯的，$n = |S|$是状态元素的个数。

1.3 无向网和网拓扑

1.3.1 无向网

二元组(X,P)称为**无向网**，如果$X \neq \varnothing \wedge P \subseteq X \times X \wedge \mathrm{dom}(P) \cup \mathrm{cod}(P) = X \wedge \mathrm{dom}(P) \cap \mathrm{cod}(P) = \varnothing$。

1.3.2 网拓扑

1. 网拓扑概念

(X,θ)构成**网拓扑**的条件是：$X \neq \varnothing \wedge \theta \subseteq \mathscr{P}(X) \wedge \forall A \subseteq \theta: \bigcup_{a \in A} a \in \theta \wedge \forall A \subseteq \theta: \bigcap_{a \in A} a \in \theta \wedge \forall x \in X: \dot{x} \in \theta \oplus \bar{\dot{x}} \in \theta$，其中$\mathscr{P}(X)$为$X$的幂集合，$\dot{x} = \{x\}$，$\bar{\dot{x}} = X - \{x\}$。

θ中的元素称为(X,θ)的开集，开集的补集为闭集。网拓扑的单体集$\dot{x} = \{x\}$或为开集或为闭集，但不能既是开集又是闭集。

网拓扑又称 **Petri** 拓扑.

2. 有向网的网拓扑结构

与有向网$(S,T;F)$对应的网拓扑是(X,θ),其中 $X = S \cup T$,θ则由满足下述条件的 X 子集 $a \subseteq X$ 构成:

$$a \in \theta \Leftrightarrow \forall t \in T \cap a: {}^{\cdot}t \cup t^{\cdot} \subseteq a.$$

3. 无向网与网拓扑

给定无向网(X,P),可以构造它的网拓扑(X,θ)如下:

$$\forall a \subseteq X:(a \in \theta \Leftrightarrow \forall t \in \mathrm{cod}(P): P^{-1}(t) \subseteq a)$$

其中
$$P^{-1}(t) = \{x \mid (x,t) \mid \in P\}.$$

给定网拓扑(X,θ),可以构造它对应的无向网(X,P)如下:

$$\mathrm{dom}(P) = \{x \mid \dot{x} \in \theta\} \wedge \mathrm{cod}(P) = \{x \mid \bar{\dot{x}} \in \theta\},$$
$$\wedge\ P = \{(x,y) \mid \forall a \in \theta;\quad y \in a \Rightarrow x \in a\}.$$

4. 有向网与无向网

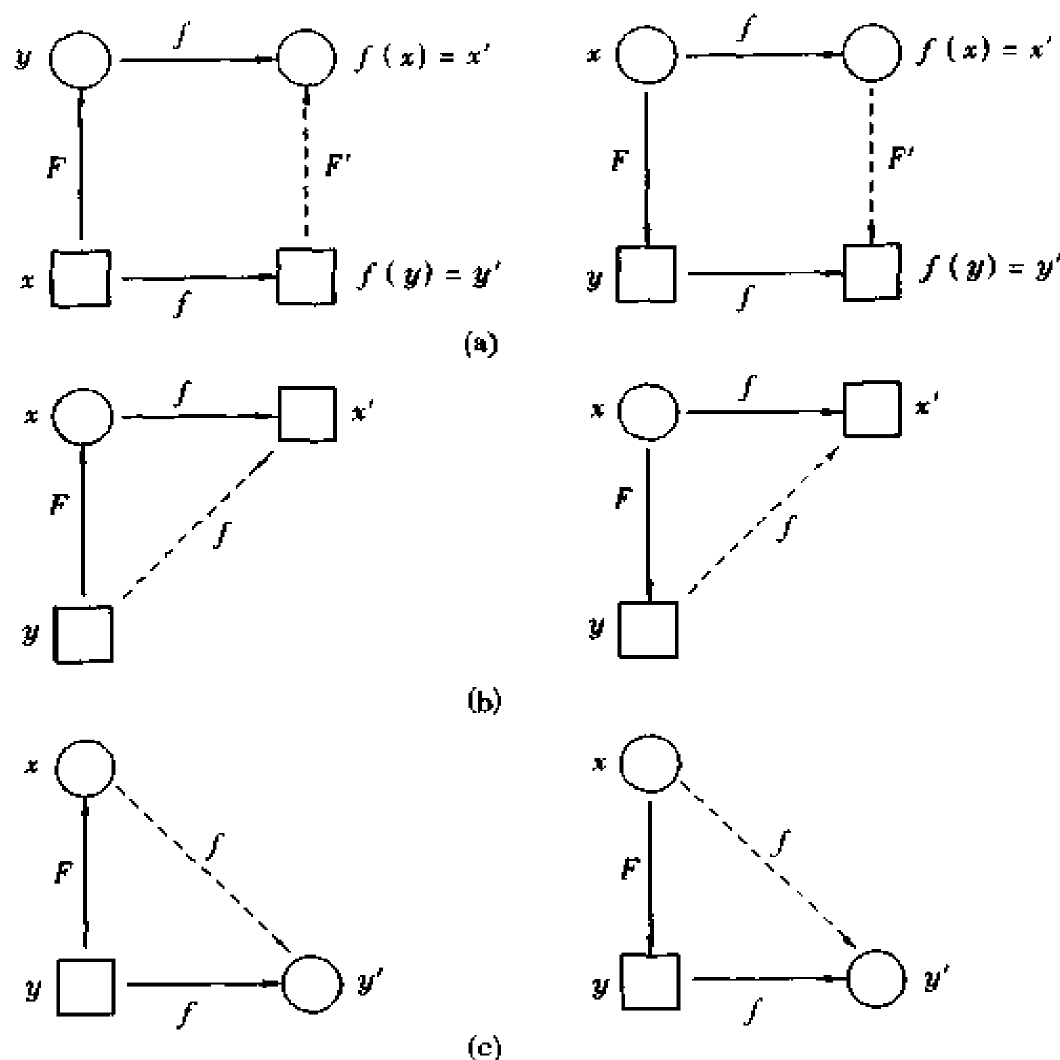


图 1-3

对给定的有向网$(S,T;F)$,令 $P=(F\cup F^{-1})\cap(S\times T)$,$(X,P)$为$(S,T;F)$决定的无向网,其中 $F^{-1}=\{(x,y)\mid(y,x)\in F\}$.

1.3.3　连续映射和网射

令(X,θ)和(X,P)为互相对应的网拓扑和无向网,(X',θ')和(X',P')为另一对网拓扑和无向网.$f:(X,\theta)\to(X',\theta')$为网拓扑下的连续映射的条件是:$\forall a'\in\theta',f^{-1}(a')\in\theta$.用无向网术语表述则是:$f:(X,P)\to(X',P')$为连续映射的条件是:$f$保$P\cup Id$,其中 Id 为恒同映射.所谓保 $P\cup Id$,是指,$\forall(x,y)\in P,f(x)=f(y)\vee(f(x),f(y))\in P'$.

令$(S,T;F)$和$(S',T';F')$为有向网,(X,P)和(X',P')分别为它们对应的无向网,$f:(S,T;F)\to(S',T';F')$为网射的条件是:f既保 $F\cup Id$,又保 $P\cup Id$.

用有向网的图形表示检查有向网之间的映射是否为网射,则 f 为网射的充分必要条件是,f 只允许图 1-3 给出的六种情况,图中实线给出的是已知条件,虚线指出 f 为连续映射时必须有的映射关系或流关系.

只允许图 1-3(a) 中对应关系的网射称为折叠.

2　网　系　统

为网结构规定变迁规则,就得到具有动态行为的网系统.

2.1　变迁规则

2.1.1　发生权

令$(S,T;F)$为有向网,K、W、M 依次为网上的容量函数、权函数和标识,$t\in T$为任一变迁,则 t 在 M 有发生权的充分必要条件是:

$$\forall s\in{}^{\cdot}t:M(s)\geqslant W(s,t)\wedge\forall s\in t^{\cdot}:M(s)+W(t,s)\leqslant K(s).$$

t 在 M 有发生权的事实记作 $M[t\rangle$. $M[t\rangle$ 也说成 t 在 M 受权发生,M 授权 t 发生.

2.1.2　后继标识

若 M 授权 t 发生,则 t 发生的结果是将标识 M 改变为如下计算的标识 M',M' 称为 M 的后继,写作 $M\xrightarrow{t}M'$ 或 $M[t\rangle M'$.

M' 的计算公式为,$\forall s\in S$,

$$M'(s)=\begin{cases}M(s)-W(s,t), & \text{若 } s\in{}^{\cdot}t-t^{\cdot},\\ M(s)+W(t,s), & \text{若 } s\in t^{\cdot}-{}^{\cdot}t,\\ M(s)+W(t,s)-W(s,t), & \text{若 } s\in{}^{\cdot}t\cap t^{\cdot},\\ M(s), & \text{其它}.\end{cases}$$

若 W 为广义权函数，则上述公式简化为

$$M'(s) = M(s) + W(t,s) - W(s,t).$$

2.2 基本网系统

2.2.1 基本网系统概念

若容量函数 $K = 1$，即 $\forall s \in S: K(s) = 1$，权函数 $W = 1$，即 $\forall (x,y) \in F: W(x,y) = 1$，则由有向网$(S,T;F)$产生的网系统为**基本网系统(EN系统)**. 此时每个状态元素只有有托肯和无托肯两种状态，可以看作成真与否的逻辑变量或条件. 与条件相关的变迁称为事件. 通常用 B 代替 S 表示条件集，用 E 代替 T 表示事件集. B 上的标识则用含托肯的那些条件所组成的子集表示. 容量函数 K 和权函数则用缺省蕴涵表示.

$\mathcal{N} = (B,E;F,c_{in})$ 称为基本网系统的条件是：$(B,E;F)$ 为有向网 $\wedge\ c_{in} \subseteq B$.

条件集合 B 的子集称为**条件丛**，简称丛. 事件 $e \in E$ 在条件丛 $c \subseteq B$ 有发生权的条件是：${}^{\bullet}e \subseteq c \wedge e^{\bullet} \cap c = \varnothing$，这是网上变迁规则在 $K = 1$ 和 $W = 1$ 时的集合符号写法. e 在 c 有发生权记作 $c[e\rangle$. $c' = (c - {}^{\bullet}e) \cup e^{\bullet}$ 称为 c 的**后继丛**，记作 $c[e\rangle c'$.

基本网系统 $\mathcal{N} = (B,E;F,c_{in})$ 的状态丛称为它的**情态**，c 为其初始情态. $\mathcal{N} = (B,E;F,c_{in})$ 的情态集 C 是满足下列条件的最小集合：

$$c_{in} \in C \wedge ((c \in C \wedge \exists e \in E: c[e\rangle c') \Rightarrow c' \in C).$$

2.2.2 独立事件集和一步发生权

谓词 Ind(u) 的定义是：Ind(u):$\Leftrightarrow u \subseteq E \wedge (\forall e_1、e_2 \in u, e_1 \neq e_2 \Rightarrow ({}^{\bullet}e_1 \cup e_1{}^{\bullet}) \cap ({}^{\bullet}e_2 \cup e_1{}^{2}) = \varnothing)$. 若 Ind($u$) 成真，就说 u 是**独立事件集**.

$u \subseteq E$，在条件丛 c 有一步发生权的条件是：Ind(u) $\wedge\ {}^{\bullet}u \subseteq c \wedge u^{\bullet} \cap c = \varnothing$，其中 ${}^{\bullet}u = \bigcup_{e \in u} {}^{\bullet}e$，$u^{\bullet} = \bigcup_{e \in u} e^{\bullet}$ 分别称为 u 的前集和后集. 一步发生权记作 $c[u\rangle$，$c' = (c - {}^{\bullet}u) \cup u^{\bullet}$ 称为 c 的一步后继，记作 $c[u\rangle c'$.

若 $c[u\rangle c'$，u_1、u_2 是 u 的一种分割，即 $u_1 \subseteq u \wedge u_2 \subseteq u \wedge u_1 \cup u_2 = u \wedge u_1 \cap u_2 = \varnothing$，则存在 c''，使得 $c[u_1\rangle c'' \wedge c''[u_2\rangle c'$，合写为 $c[u_1\rangle c''[u_2\rangle c'$.

2.2.3 情态可达图

以基本网系统 $\mathcal{N} = (B,E;F,c_{in})$ 的情态为节点，情态间的后继关系用有向弧表示，就得到 $\mathcal{N}$ 的**情态可达图**，弧上是发生的事件.

例1 $\mathcal{N} = (\{b_1,b_2,b_3\},\{e_1,e_2,e_3\};\{(b_1,e_1),(e_1,b_2),(b_2,e_2),(e_2,b_1),(b_1,e_3),(e_3,b_3)\},\{b_1\})$ 的图形表示如图2-1所示，图2-2则是它的情态可达图，$\mathcal{N}$ 的情态集 $C = \{\{b_1\},\{b_2\},\{b_3\}\}$.

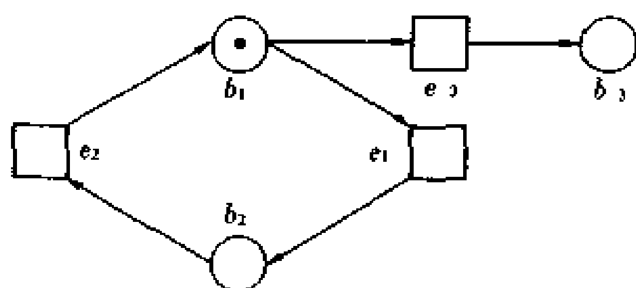


图 2-1

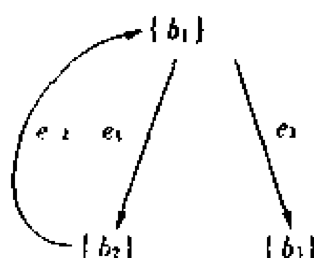


图 2-2

2.2.4　事件间的基本关系

(1) 顺序关系. 若 $c[e_1\rangle c'[e_2\rangle$, 但 $\neg\ c[e_2\rangle$, 就说 e_1 和 e_2 在 c 有顺序关系.

(2) 冲突. 若 $c[e_1\rangle \wedge c[e_2\rangle$, 但 $\neg\ c[\{e_1, e_2\}\rangle$, 就说 e_1 和 e_2 在 c 互相冲突.

例 2　图 2-3(a) 和(b) 中的 e_1 和 e_2 在图中托肯给出的情况下均互相冲突.

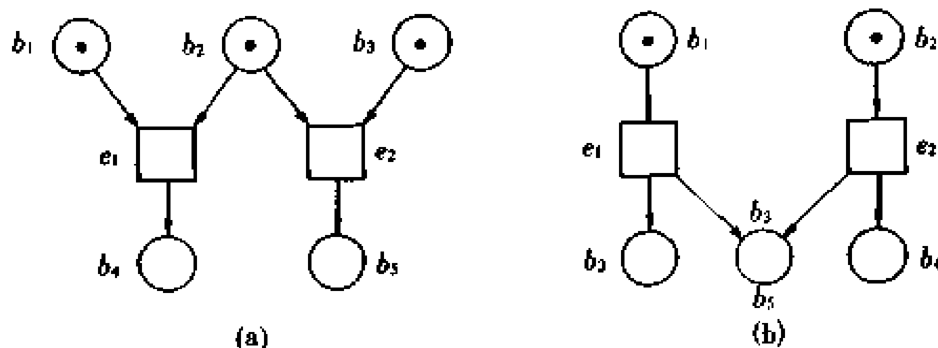


图 2-3

(3) 冲撞　若对 $b \in B$ 有 $e \in E$ 和 $c \in C$, 使 $^{\cdot}e \subseteq c \wedge b \in (c \cap e^{\cdot})$, 就说在情态 c 条件 b 处有冲撞.

(4) 并发.　若 $c[\{e_1, e_2\}\rangle$, 就说 e_1 和 e_2 在 c 并发, 此时 $^{\cdot}e_1 \cup {^{\cdot}e_2} \subseteq c \wedge (e_1^{\cdot} \cup e_2^{\cdot}) \cap c = \varnothing \wedge \mathrm{Ind}(\{e_1, e_2\})$.

若在任何情态下任何条件处都没有冲撞, 就说 $\mathcal{N}$ 为无冲撞系统. 对 $(B, E; F)$ 作 S- 补操作可以消除冲撞.

2.3　库所 / 变迁系统

若有向网上流动的是物质资源, 状态元素就是盛放资源的地方. 用不同的状态元素盛放不同类的资源, 就得到库所 / 变迁系统. 之所以把状态元素称为库所, 是因为一方面它类似仓库, 有一定的容量, 能容纳一定种类的资源; 另一方面它又不同于仓库, 没有位置属性.

2.3.1　库所 / 变迁系统概念

$\Sigma = (S, T; F, K, W, M_0)$ 称为库所 / 变迁系统的条件是: $(S, T; F)$ 为有向网,

K、W 和 M_0 依次是$(S,T;F)$上的容量函数、权函数和标识.M_0 为 Σ 的初始标识,S 是 Σ 的库所集.库所/变迁系统简称为 **P/T 系统**.

当容量函数 $K=\omega$,权函数 $W=1$时,P/T系统称为 **P/T网**.这是Petri最早使用的网系统.

例3 图2-4是用图形方式给出的一个P/T系统,这是生产流水线上截取的两个生产环节,其中 s_1,s_2,s_3 表示半成品,s_4,s_6 表示待组装部件,s_5 表示螺丝钉,s_0 表示工具,t_1,t_2 表示组装活动.

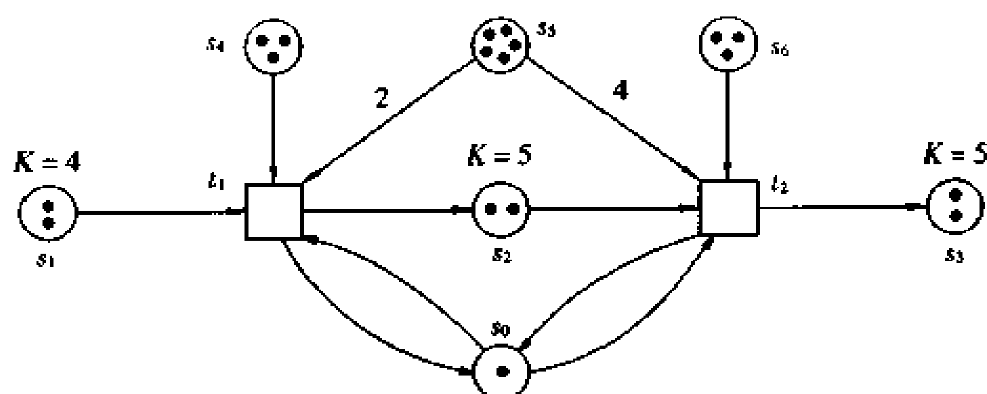


图 2-4

P/T-系统的动态行为由变迁规则界定,可以用可达树、可达图、变迁序列和进程等方式描述.

以下总假定库所集 $S=\{s_2,s_2,\cdots,s_n\}$,变迁集 $T=\{t_1,t_2,\cdots,t_m\}$.

2.3.2 可达树和可达图

映射 $M:S\to N\cup\{\omega\}$ 称为库所集 S 上的标记.标记与标识的区别在于前者可以以 ω 为值.对标识定义的发生权和后继可以照搬到标记上,不再重复.其实标识是标记的特例.

令 M_1 和 M_2 均为标记,若 $\forall s\in S:M_1(s)\leqslant M_2(s)$,就说 M_2 覆盖 M_1,记作 $M_1\leqslant M_2$.若 $M_1\leqslant M_2$ 但 $M_1\neq M_2$,就说 $M_1<M_2$.此时若 $M_1(s)<M_2(s)$,就说 $M_1<M_2$ 在 s 成立.

由下述算法构造的树结构称为 $\Sigma=(S,T;F,K,W,M_0)$ 的**可达树**,树的每个节点均带有 S 上的标记:节点 x 的标记记作 M_x.构造算法为:

(1) 可达树变量 $T(\Sigma)$ 初值只有根节点 r,且 $M_r=M_0$,即 r 以 M_0 为标记.

(2) 任取 $T(\Sigma)$ 的叶节点 x,按下述标准检查 x 是否为真叶节点:若 M_x 不授权任何变迁发生,x 为真叶节点;若从 r 到 x 的有向路径上有节点 y,$y\neq x$ 但 $M_y=M_x$,则 x 也为真叶节点.

若 $T(\Sigma)$ 的叶节点均为真叶节点,构造完毕.否则如下处理叶节点 x.

(3) 对 M_x 授权发生的每个变迁 $t\in T$,在 $T(\Sigma)$ 上增加 x 的一个子节点 y,从 x 到 y 的有向弧以 t 标记,y 的标记 M_y 分两步计算:先计算 M_x 的后继 M':$M_x[t\rangle M'$,再检查从根节点 r 到 x 的路径上所有节点标记,若没有 z 使 $M_z<M'$,则 $M_y=M'$,

否则对 $s \in S$，

$$M_y(s) = \begin{cases} M'(s), & 若\ M_z(s) = M'(s), \\ \omega, & 若\ M_z(s) < M'(s). \end{cases}$$

(4) 重复步骤(2).

将可达树 $T(\Sigma)$ 上标记相同的节点合并为一个就得到 Σ 的**可达图** $G(\Sigma)$.

定理 1 $T(\Sigma)$ 为有限树，即可达树构造算法能终止.

2.3.3 可达标识集

Σ 的可达标识集 $[M_0\rangle$ 是满足下列条件的最小集合：

1° $M_0 \in [M_0\rangle$.

2° 若 $M \in [M_0\rangle$，且 $M[t\rangle M'$，其中 $t \in T$，则 $M' \in [M_0\rangle$.

定理 2 $T(\Sigma)$ 和 $[M_0\rangle$ 的关系

1° 若 $M \in [M_0\rangle$，则有节点 $x \in T(\Sigma)$，使 $M \leqslant M_x$，且若 M_x 不以 ω 为值时 $M = M_x$.

2° 令 M_x 是 $T(\Sigma)$ 上的节点标记，则若 M_x 不以 ω 为值，必有 $M_x \in [M_0\rangle$；若有 $s \in S$ 使得 $M_x(s) = \omega$，则对任意正整数 $i > 0$，必有 $M \in [M_0\rangle$，使 $M(s) > i$，且 $\forall s' \in S, M_x(s') \neq \omega \Rightarrow M(s') = M_x(s')$.

2.3.4 变迁序列

$\sigma = M_0 t_1 M_1 t_2 M_2 \cdots t_n M_n$ 称为 **Σ 的发生序列**，如果 $\forall i, 1 \leqslant i \leqslant n, M_{i-1}[t_i\rangle M_i$，$n$ 称为 σ 的长度. 无限序列 $\sigma = M_0 t_1 M_1 t_2 M_2 \cdots$ 若满足 $\forall i, 0 < i, M_{i-1}[t_i\rangle M_i$ 就称为 Σ 的无限发生序列.

若 $\sigma = M_0 t_1 M_1 t_2 M_2 \cdots t_n M_n$ 为长度等于 n 的发生序列，则 $\tau = t_1 t_2 \cdots t_n$ 为长度等于 n 的**变迁序列**. 若 $\sigma = M_0 t_1 M_1 t_2 M_2 \cdots$ 为无穷发生序列，则 $\tau = t_1 t_2 \cdots$ 为无穷变迁序列.

令 T_1、$T_2 \subseteq T$ 为任意非空变迁子集，若存在无穷发生序列 σ，使得 $\#(\sigma, T_1) < \infty$ 而 $\#(\sigma, T_2) = \infty$ 或 $\#(\sigma, T_1) = \infty$ 而 $\#(\sigma, T_2) < \infty$，就说 Σ 对 T_1、T_2 是不公平的，其中 $\#(\sigma, T_i), i = 1,2$，为 T_i 中事件出现在序列 σ 中的次数. 这时也说 Σ 有饥饿存在. 总对任何非空 T_1、$T_2 \subseteq T$，Σ 都不是不公平的，就说 Σ 是**公平系统**.

2.3.5 Σ 的活性

$t \in T$ 是 **Σ 的活变迁**的条件是：$\forall M \in [M_0\rangle, \exists M' \in [M\rangle : M'[t\rangle$. 换言之，若对 Σ 的任何可达标识 $M \in [M_0\rangle$，都有 M 可达的标识 $M' \in [M\rangle$，t 在 M' 有发生权，则 t 就是活的.

若 Σ 的所有变迁都是活的，Σ 就是活的.

2.3.6 Σ 的进程

有向网 $(B, E; F)$ 为**出现网**的条件是：$(\forall b \in B, |{}^{\cdot}b| \leqslant 1 \wedge |b^{\cdot}| \leqslant 1) \wedge$

$F^{+} \cap (F^{-1})^{+} = \varnothing$,其中 $F^{+} = \bigcup_{i=1}^{\infty} F^{i}$,$(F^{-1})^{+} = \bigcup_{i=1}^{\infty} (F^{-1})^{i}$,$F^{1} = F$,$F^{2} = F \cdot F$,…,$F^{n} = F^{n-1} \cdot F \quad (n > 1)$.

令 $N = (B, E; F')$ 为出现网,$\Sigma = (S, T; F, K, W, M_0)$ 为 P/T- 系统,$\rho: N \to \Sigma$,(N, ρ) 称为 Σ 上的一个进程,如果

$$\begin{aligned}
&\rho(B) \subseteq S \wedge \rho(E) \subseteq T \wedge \forall (x, y) \in F': \rho(x, y) = (\rho(x), \rho(y)) \in F \\
&\wedge \forall e \in E: \rho({}^{\cdot}e) = {}^{\cdot}\rho(e) \wedge \rho(e^{\cdot}) = \rho(e)^{\cdot} \\
&\wedge \forall b_1 : b_2 \in B: \rho(b_1) = \rho(b_2) \wedge b_1 \neq b_2 \Rightarrow {}^{\cdot}b_1 \neq {}^{\cdot}b_2 \wedge b_1^{\cdot} \neq b_2^{\cdot} \\
&\wedge \forall s \in S: |\{b \mid {}^{\cdot}b = \varnothing \wedge \rho(b) = s\}| \leqslant M_0(s).
\end{aligned}$$

2.3.7　不变量

令 C 为 P/T 系统 Σ 的关联矩阵,$T = \{t_1, t_2, \cdots, t_m\}$ 为 Σ 的变迁集,σ 为 Σ 的有限发生序列,M 为 σ 的最终标识,则

$$M = M_0 + C \cdot u,$$

其中 u 为 m 维列向量,它的分量 u_i,$i = 1, 2, \cdots, m$,是变迁 t_i 在 σ 出现的次数,即 $u_i = \#(\sigma, t_i)$.

令 $x_1, x_2, \cdots, x_m$ 为变量,X 是以 x_i 为第 i 分量的 m 维列向量,θ_s 为分量全为0的 n 维列向量,则齐次线性方程组 $CX = \theta_s$ 的非负整数列向量解称为 Σ 的 **T 不变量**.

令 $x_1, x_2, \cdots, x_n$ 为变量,X 是以 x_i 为第 i 分量的 n 维列向量,θ_T 为分量全为0的 m 维列向量,C^{T} 是关联矩阵 C 的转置矩阵,则齐次线性方程组 $C^{T}X = \theta_T$ 的非负整数解称为 Σ 的 **S 不变量**.

2.3.8　基本现象

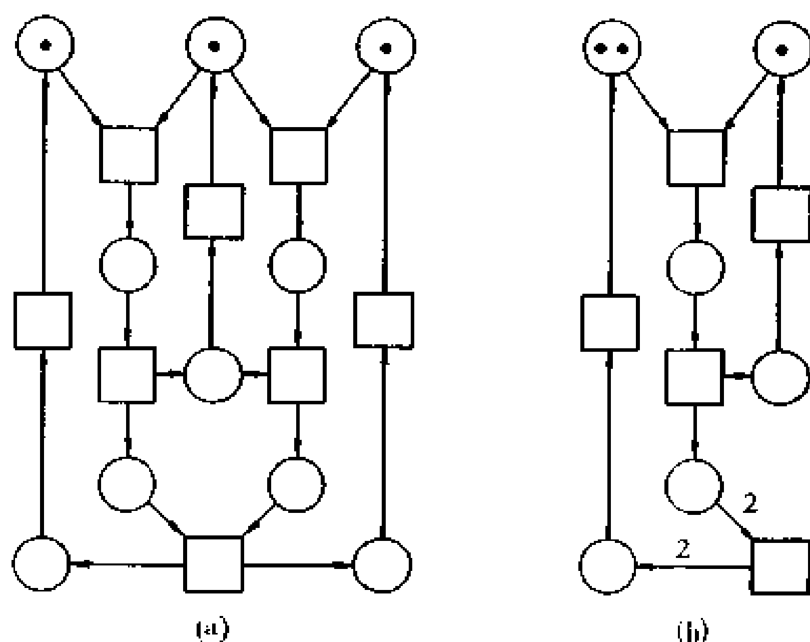
P/T- 系统的基本现象与基本网系统类似:由共享资源的传递带来顺序关系,对共享资源的竞争引来冲突,对空间的竞争产生冲撞,互相独立的变迁则并发.

2.4　高级网系统

2.4.1　高级网系统概念

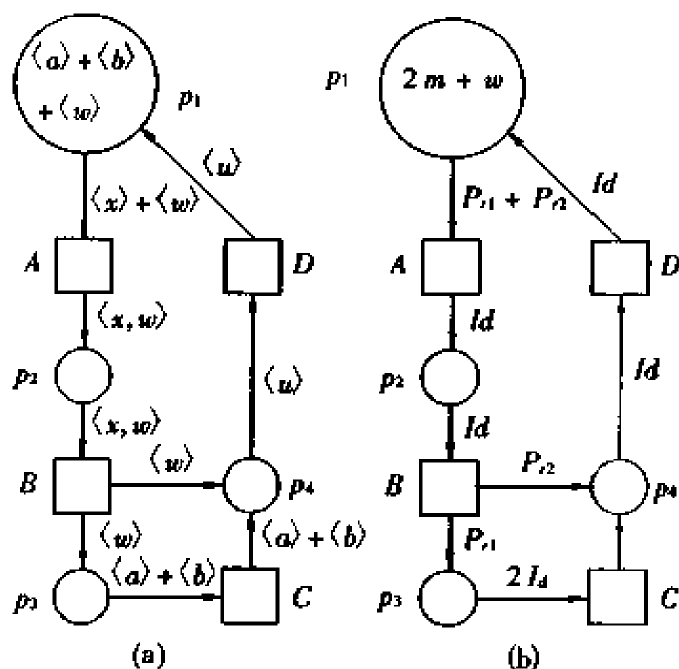
为减少网系统中节点个数,可以把作用相同的不同类资源用同一个状态元素表示,资源的种类则以个体名或不同染色表示.这时状态元素内资源不再是不可区分的托肯,而是有个性的托肯,因而这类网系统又称**个性托肯系统**.若资源用个体名区分,作用相同的不同个体的状态用谓词表示,就得到**谓词/变迁系统**,简写为 **Pr/T 系统**.若资源用染色来区别种类,则为**有色网系统**.当然,在上述过程中,作用相同的变迁也要作相应合并.这种从网到网的变化是网射.个性托肯系统、谓词/变迁系统、有色网系统都是高级网系统.

例 4　图 2-5(a) 是个基本网系统,显然它左右对称,合并左右得到图 2-5(b)



(a)　　(b)

图 2-5



(a)　　(b)

图 2-6

中的 P/T- 系统. 图 2-6(a) 是进一步合并成的 Pr/T 系统, 其中 a 和 b 是图 2-5(a) 左右两边的资源名, w 是中间的资源名. $\langle a\rangle+\langle b\rangle+\langle w\rangle$ 称为符号和, 表示三个独立个体. x 是变量名, $\langle x,w\rangle$ 是二元符号和, 是由 x 当前代表的个体和个体 w 结合在一起的共存形式. u 也是变量名. 图 2-6(b) 中用字母 m 表示 a 和 b 共同的颜色, w 仍用同一字母表示它的颜色, $2m+w$ 是表示 2 个 m 色和 1 个 w 色资源的多重集表达式. $\langle m,w\rangle$ 则是 m 色和 w 色各一个资源结合在一起的复合色资源.

2.4.2 Pr/T 系统

活动于网系统上的资源个体集合称为个体集或个体名集，通常用 D 表示 . D 上的变量集合用 V 表示 . 若 $f^{(n)}: D^n \to D$ 为 D 上 n 元运算符，则 $d \in D, v \in V$ 称为 D 上的项，$f^{(n)}(v_1, v_2, \cdots, v_n)$ 也称为 D 上的项，只要其中 $v_1, v_2, \cdots, v_n$ 为 D 上的项.

以 D 上的项为分量的 n 元向量 $(v_1, v_2, \cdots, v_n)$ 称为 D 上的 n 元组，$n \geqslant 1$. 用加号"+"连接 n 元组构成的形式和称为 D 上的 n 元符号和 .

D 上的公式有以下几种形式：v_1、v_2 为 D 上的项，$v_1 = v_2$ 为公式；若 p 为公式，则 $\neg p$ 也是公式；若 p、q 为公式，则 $p \vee q$ 也是公式，$(\exists x)p$ 也是公式，其中 x 是 D 上变量 .

若谓词 p 的主体为 D 上 n 元组，则 p 称为 D 的 n 元谓词 . $p(D) = \{(d_1, d_2, \cdots, d_n) \mid p(d_1, d_2, \cdots, d_n)\}$ 称为 p 的外延 . 若 $p(D)$ 是不变的，p 称为静态谓词，若 $p(D)$ 随网系统的动态变化而变化，p 称为可变谓词 .

$\Sigma = (P, T; F, D, V, A_P, A_T, A_F, M_0)$ 构成 Pr/T 系统的条件是：

1° $(P, T; F)$ 为有限网，称为 Σ 的基网，

2° D 为非空有限集，称为 Σ 的个体集；Ω 为 D 上已知的运算符集，

3° V 为 D 上的变量集，

4° $A_P: P \to \pi$，其中 π 是 D 上的可变谓词集 . 对 $p \in P$ 若 $A_P(p)$ 是 n 元谓词，就说 p 是 n 元谓词 .

5° $A_T: T \to f_D$，其中 f_D 是 D 的公式集 . 对 $t \in T$，$A_T(t)$ 只能包含静态谓词和 Ω 中运算符 .

6° $A_F: F \to f_s$，其中 f_s 是 D 的符号和集，使得对 n 元谓词 $p \in P$，若 $(t, p) \in F$ 或 $(p, t) \in F$，则 $A(t, p)$ 或 $A(p, t)$ 为 n 元符号和 . 对于 $t \in T$，出现在 $A_T(t)$ 中的自由变量恰为出现在 $A_F(p, t)$ 或 $A_F(t, p)$ 中的自由变量 .

7° $M_0: P \to f_s$，使得对 n 元谓词 $p \in P$，$M_0(p)$ 为 n 元符号和，使每个个体 $d \in D$ 恰属于一个 $M_0(p)$.

例 5 图2-6(a) 中 p_1、p_2、p_3 和 p_4 依次为 1 元、2 元、1 元和 1 元谓词，其含义分别为就绪(ready)、工作(working)、等待(waiting) 和休息(resting). 这是把个体 a 和 b 理解为机床，w 为工人，从而整个网系统描述的是 w 交替开动(变迁 A)a 和 b、结束(变迁 B)、组装(变迁 C) 和恢复(变迁 D) 的过程. 结束动作使 w 处于休息状态，机床 a 或 b 处于等待状态. 组装则是 a 和 b 共同参与的，然后共同进入休整态(p_4). 显然，随着个体 a、b 和 w 状态的改变，p_1、p_2、p_3 和 p_4 的外延也在改变 . 图中隐涵的各项如下：

$$D = \{a, b, w\}, \quad V = \{x, u\},$$

$$A_P(p_1) = \text{ready}, \ A_P(p_2) = \text{working}, A_P(p_3) = \text{waiting}, \ A_P(p_4) = \text{resting},$$

$$A_T(A) = \text{true}, \ A_T(B) = \text{true}, A_T(C) = \text{true}, \ A_T(D) = \text{true},$$

A_F 已经在图中有向弧上明确给出 . M_0 也可从图中准确确定 .

注意 true 是 D 上的零元静态谓词．将 $A_T(A)$ 改为 $x \neq w$，其中“$\neq w$”为一元静态谓词，仍是同一系统，因为 Pr/T 系统的变迁规则使这一改动保持系统的行为．

2.4.3 Pr/T 系统的变迁规则

$M: P \to f_s$ 称为 Σ 的标识的条件是：若 $p \in P$ 为 n 元谓词，则 $M(p)$ 为 n 元符号和；对任何个体 $d \in D$，存在唯一的 $p \in P$，使 d 出现在 $M(p)$ 的唯一的一个 n 元组中．

令 f_1 和 f_2 为两个 n 元符号和，$f_1(D)$ 和 $f_2(D)$ 分别是它们所含 n 元组之集合，则 $f_1 = f_2 :\Leftrightarrow f_1(D) = f_2(D)$，$f_1 \leqslant f_2 :\Leftrightarrow f_1(D) \subseteq f_2(D)$，$f_1 < f_2 :\Leftrightarrow f_1 \leqslant f_2 \wedge f_1 \neq f_2$．$f_1 \cap f_2 :\Leftrightarrow f_1(D) \cap f_2(D)$．“$:\Leftrightarrow$”读作“定义为”．

令 $v_1, v_2, \cdots, v_n$ 为变迁 $t \in T$ 上的自由变量，即出现在 $A_T(t)$、$A_F(p,t)$ 和 $A_F(t,p)$ 中的自由变量，用个体 $d_1, d_2, \cdots, d_n \in D$ 依次替换 $v_1, v_2, \cdots, v_n$，记作 $(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)$，称为 t 的变量替换．实施这一替换后的 $A_T(t)$、$A_F(p,t)$ 和 $A_F(t,p)$ 分别写作 $A_T(t)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)$，$A_F(p,t)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)$，$A_F(t,p)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)$．如果这些谓词实例满足下述条件，$(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)$ 就是标识 M 下的可行替换：$A_T(t)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n) = \text{true} \wedge M(p) \geqslant A_F(p,t)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n) \wedge M(p) \cap A_F(t,p)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n) = \varnothing$．$M$ 下的可行替换记作 $M[t(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)\rangle$．

若 $M[t(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)\rangle$，就说变量替换 $t(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)$ 在 M 有发生权．

若 $M[t(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)\rangle$，则 M 的后继标识 M' 如下计算．对 $p \in P$，

$$M'(p) = M(p) - A_F(p,t)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n) + A_F(t,p)(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n),$$

其中符号和减法是从前一符号和中去掉后者所含的那些 n 元组．后继关系记作：

$$M[t(v_1 \leftarrow d_1, v_2 \leftarrow d_2, \cdots, v_n \leftarrow d_n)\rangle M'$$

Pr/T 系统要求它的标识是 D 中个体的一种分布，但并不排斥系统与其外界环境的交流．在 P 中引入 in 和 out 两个谓词，即可处理进入和退出系统的个体，以始终保持这些个体的踪迹．

2.4.4 Pr/T 系统行为

其可达标识集是由 M_0 出发经有限步变迁发生所得后继标识的集合，记作 $[M_0\rangle$．

其可达树与 P/T 系统可达树类似，只是不会有 ω 出现．Pr/T 系统可达标识集与可达树节点标记集合相等．合并可达树上标记相同的节点得到可达图．

其发生序列和变迁序列与 P/T 系统的相应序列的区别是，序列中不是变迁名，而是变迁的可行替换．所以变迁序列应改称可行替换序列．

Pr/T系统的进程同样是出现网 $N=(B,E;F')$ 和出现网到系统基网$(S,T;F)$的映射 ρ，只是对 $e\in E$，$\rho(e)$ 应该是 T 中变迁的可行替换．

Pr/T 系统中也有冲突、顺序与并发，但没有冲撞，因为变迁的发生只是改变个体的状态，不涉及占用空间的问题．

2.4.5 Pr/T 系统的不变量

1. 关联矩阵

$\Sigma=(P,T;F,D,V,A_P,A_T,A_F,M_0)$ 的关联矩阵 C 的矩阵元素 c_{ij}，$1\leqslant i\leqslant n$，$1\leqslant j\leqslant m$，由下式给出：

$$c_{ij}=-A_P(p_i,t_j)+A_P(t_j,p_i),$$

其中假定了 $P=\{p_1,p_2,\cdots,p_n\}$，$T=\{t_1,t_2,\cdots,t_m\}$，而且若 $p_i\notin {}^{\cdot}t_j$，则 $A_P(p_i,t_j)=\langle\ \rangle$，若 $p_i\in t_j^{\cdot}$，则 $A_P(t_j,p_i)=\langle\ \rangle$，$\langle\ \rangle$ 为空符号和．

例 6 表2-1 给出的是图 2-6(a) 中 Pr/T 系统的关联矩阵．

表2-1

	A	B	C	D
p_1	$-\langle x\rangle-\langle w\rangle$	$\langle\ \rangle$	$\langle\ \rangle$	$\langle u\rangle$
p_2	$\langle x,w\rangle$	$-\langle x,w\rangle$	$\langle\ \rangle$	$\langle\ \rangle$
p_3	$\langle\ \rangle$	$\langle x\rangle$	$-\langle a\rangle-\langle b\rangle$	$\langle\ \rangle$
p_4	$\langle\ \rangle$	$\langle w\rangle$	$\langle a\rangle+\langle b\rangle$	$-\langle u\rangle$

2. S 不变量

令 R 为分解 n 元组为 n 个一元组的可分配运算，则 n 维行向量 $u=(u_1,u_2,\cdots,u_n)$ 为 Σ 的 S 不变量的充分必要条件是：$\forall i:1\leqslant i\leqslant n$，

$$u_i=0\vee u_i=1\wedge\forall M\in[M_0\rangle:\sum_{i=1}^{n}u_iR(M(p_i))=\sum_{i=1}^{n}u_iR(M_0(p_i)).$$

3. T 不变量

m 阶列向量 u 为 Σ 的 T 不变量的条件是：u 的分量 u_j 为非负正整数，但不全为 0，而且存在标识 M 到 M 自己的可行替换序列，使 t_j 恰出现在序列中 u_j 次，$j=1,2,\cdots,m$．应该指出的是，T 不变量与初始标识 M_0 无关，定义中的标识 M 也不一定是可达标识．当 $M\in[M_0\rangle$ 时，相应的 T 不变量才与由 M_0 决定的 Pr/T系统行为有关．

2.4.6 有色网系统

1. 颜色集和复合色

颜色用来为个体分类，同类个体只计个数．于是由不同类别不同个数的一组资源用颜色集上的多重集表示．单色符号的多元组代表复合色．现用 D 表示颜色集．

2. 有色网系统定义

$\Sigma=(P,T;F,I_+,I_-,M_0)$ 称为有色网系统的条件是：

1° $(P,T;F)$ 为有向网,称为 Σ 的基网.

2° $C:P\cup T\to\mathscr{P}(D)$,使得

对所有 $p\in P$,$C(p)$ 是 p 上所有可能的托肯色之集合,即 p 能容纳的资源种类之集合.

对所有 $t\in T$,$C(t)$ 是 t 的所有出现色之集合,即 t 的发生涉及的颜色搭配.

3° I_+ 和 I_- 分别是 $P\times T$ 上的正函数和负函数.对 $(p,t)\in P\times T$,

$I_+(p,t)\in[C(t)_{MS}\to C(p)_{MS}]_L$ 且 $I_+(p,t)=0$,当且仅当 $(t,p)\notin F$,其中 $C(t)_{MS}$,$C(p)_{MS}$ 分别是 $C(t)$ 和 $C(p)$ 上的多重集,$[A\to B]_L$ 则是从 A 到 B 的线性函数的集合.

$I_-(p,t)\in[C(t)_{MS}\to C(p)_{MS}]_L$ 且 $I_-(p,t)=0$,当且仅当 $(p,t)\notin F$.

4° M_0 为 Σ 的初始标识,若 $M_0:P\to D_{MS}$,使得 $\forall p\in P,M_0(p)\in C(p)_{MS}$.

3. 有色网系统关联矩阵

以 $I_+(p_i,t_j)$ 为矩阵元素的 $n\times m$ 阶矩阵 I_+ 是 Σ 的正矩阵,以 $I_-(p_i,t_j)$ 为矩阵元素的 $n\times m$ 阶矩阵 I_- 为 Σ 的负矩阵,而 I_+-I_- 则是 Σ 的(整体)关联矩阵.注意 $P=\{p_1,p_2,\cdots,p_n\}$,$T=\{t_1,t_2,\cdots,t_m\}$.

例7 表2-2给出的是图2-6(b)中有色网系统的关联矩阵及初始标识.表中变迁名下给出的是变迁的出现色,即 $C(A)$,$C(B)$ 等,例如表中 A 的出现色为 $\{<m,w>\}$.p_1、p_2、p_3 和 p_4 旁给出的是库所的托肯色,例如 p_1 允许 m、w 两种颜色,p_2 只允许复合色 $<m,w>$.

表2-2

	A	B	C	D	M_0
	$\{\langle m,w\rangle\}$	$\{\langle m,w\rangle\}$	$\{m\}$	$\{m,w\}$	
$p_1\{m,w\}$	$-(p_r1+p_r2)$	0	0	ID	$2m+w$
$p_2\{\langle m,w\rangle\}$	ID	$-ID$	0	0	0
$p_3\{m\}$	0	P_r1	$-2\cdot ID$	0	0
$p_4\{m,w\}$	0	p_r2	$2\cdot ID$	$-ID$	0

4. 变迁规则:一步发生权

(1) 有色网系统 $\Sigma=(P,T;F,C,I_+,I_-,M_0)$ 的标识 M 是定义在 P 上的函数,使得 $\forall p\in P:M(p)\in C(p)_{MS}$.

(2) 对 $\sum$ 的标识 M,若定义在 T 上的函数 X 使得 $\forall t\in T:X(t)\in C(t)_{MS}\wedge\forall p\in P:\sum_{t\in T}I_-(p,t)(X(t))\leqslant M(p)$,就说 X 在 M 下有一步发生权.

(3) 若 X 在 M 有一步发生权,则 X 发生产生的后继标识 M' 由下式给出:$\forall p\in P$,

$$M'(p)=M(p)+\sum_{t\in T}I_+(p,t)(X(t))-\sum_{t\in T}I_-(p,t)(X(t))$$

5. 不变量

(1) S 不变量.n 阶行向量 $(f_1,f_2,\cdots,f_n)$ 称为 Σ 的 S 不变量的条件是:$f_i\in$

$[C(p_i)_{MS} \to D'_{MS}]_L, i = 1,2,\cdots,n$，其中 D' 是单色颜色集，而且 $(f_1,f_2,\cdots,f_n) * I_- = (f_1,f_2,\cdots,f_n) * I_+$，其中 I_- 和 I_+ 分别为 Σ 的负矩阵和正矩阵．

(2) T 不变量．m 阶列向量 $(v_1,v_2,\cdots,v_m)^T$ 为 Σ 的 T 不变量的条件是：$v_j \in C(t_j)_{MS}, j = 1,2,\cdots,m$，而且 $I_- * (v_1,v_2,\cdots,v_m)^T = I_+ * (v_1,v_2,\cdots,v_m)^T$，其中 I_- 和 I_+ 为 Σ 的负矩阵和正矩阵．

2.5 自控网系统

2.5.1 自控网系统概念

如果允许变迁的发生对某些类资源数量上的改变不是常数，而是随系统状态不同而改变，就得到有自控能力的**自控网系统**(cyber 系统)．这是非线性系统．

1. 自控网系统定义

有向网 $(S,T;F)$，$W: F \to N_0 \cup S$ 称为该网的**可变权函数**，若 $W(x,y) = s, s \in S$，则 (x,y) 上的权随库所 s 中托肯数的改变而改变．

$\Sigma = (S,T;F,K,W,M_0)$ 为自控网系统的条件是：$(S,T;F)$ 为有向网，K 是它的容量函数，W 是它的可变权函数，M_0 为初始标识．

例 8 在图2-7给出的自控网系统中，$W(x,y) = s$ 是用从库所 s 到有向弧 (x,y) 的以小圆圈为箭头的弧线表示的，例如 $W(t_1,s_3) = s_4$．图中 $K = \omega$，权缺省的均为 1．

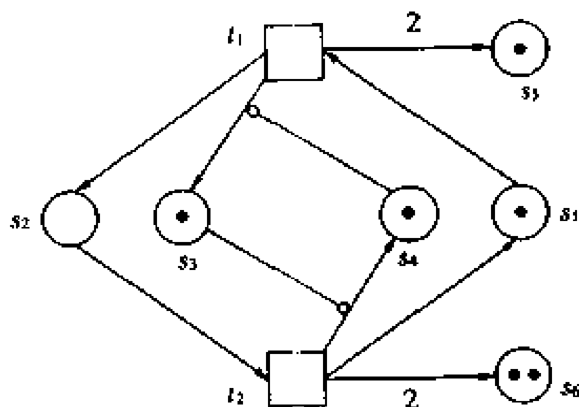


图 2-7

2. 变迁规则

自控网系统的标识与 P/T 系统一样，均为 $M: S \to N_0$，即为每个库所指明所含的资源(托肯)个数．就每个库所代表一类资源，其中托肯无个性区分而言，自控网系统属于 P/T 系统这一层次．但自控网系统是非线性 P/T 系统，因为变迁对资源数量的依赖是非线性的．

(1) 权函数当前值．对给定的标识 M，若 $W(x,y) = s, s \in S$，则 $W(x,y)$ 在 M 下的当前值为 $W(x,y) = M(s)$．若 $W(x,y) = n, n \in N$，则在任何标识下 $W(x,y)$ 的当前值均为 n．M 下的当前值记作 $W_M(x,y)$．

(2) 发生权．变迁 $t \in T$ 在标识 M 有发生权的条件是：$\forall s \in {}^{\cdot}t: M(s) \geqslant W_M(s,t) \wedge \forall s \in t^{\cdot}: M(s) + W_M(t,s) \leqslant K(s)$．发生权记作 $M[t>$．

(3) 后继标识．若 $M[t>$，则 t 发生后的后继标识 M' 由下式给出：$\forall s \in S$，

$$M'(s) = M(s) + W_M(t,s) - W_M(s,t),$$

记作 $M[t> M'$．注意 W 为广义权函数．

3. 关联矩阵

若 $S = \{s_1, s_2, \cdots, s_n\}$, $T = \{t_1, t_2, \cdots, t_m\}$，则 Σ 的关联矩阵 C 为 $n \times m$ 阶矩阵，其元素 $c_{ij} = W(t_j, s_i) - W(s_i, t_j)$.

由于矩阵元素可能为库所名，而库所所含托肯数不是常数，所以关联矩阵不是常数矩阵，自控系统也就不是线性解．

2.5.2 自控系统行为

1. 并发权

变迁 t、$t' \in T$ 在标识 M 有并发权的条件是：$\forall s \in S: (M(s) \geq W_M(s,t) + W_M(s,t') \wedge K(s) \leq M(s) + W_M(t,s) + W_M(t',s))$. 并发权记作 $M[\{t,t'\}\rangle$. t 和 t' 并发的后继 M' 由下式给出：$\forall s \in S$,

$$M'(s) = M(s) - W_M(s,t) - W_M(s,t') + W_M(t,s) + W_M(t,s').$$

2. 可达标识集

若 $M[\{t,t'\}\rangle$，且 $M[\{t,t'\}\rangle M'$, $M[t\rangle M_1'[t'\rangle M''_1$ 及 $M[t'\rangle M'_2[t\rangle M''_2$，则对P/T系统 $M' = M''_1 = M''_2$，而对自控网系统，M'、M_1'' 和 M_2'' 有可能是三个不同的标识．因而自控网系统的可达标识集是由 M_0 出发经变迁所有可能的顺序及并发能到达的标识之集合．

3. 可达树

在P/T系统可达树构造算法中，对非真叶节点 x 不仅要对 M_x 授权的每个变迁添加 x 的一个子节点，而且要对 M_x 授权并发的每一组变迁添加一个子节点．

2.5.3 不变量

1. 替换加运算

n 阶列向量 A 和 $n \times k$ 阶矩阵 Y 之间的运算 $A \leftrightarrow Y$ 定义如下($k \geq 1$)：

1° A 的分量 $a_1, a_2, \cdots, a_n$ 为常量；

2° Y 中允许出现与 $a_1, a_2, \cdots, a_n$ 对应的变量 $v_1, v_2, \cdots, v_n$；

3° 当 $k = 1$ 时 $r = A \leftrightarrow Y = A + Y|_A$，其中 $Y|_A$ 是将出现在 Y 中的变量 $v_1, v_2, \cdots, v_n$ 用 A 的相应分量 $a_1, a_2, \cdots, a_n$ 替换得到的 n 阶列向量；

4° 当 $k > 1$ 时，令 $r_1 = A \leftrightarrow Y_1$，其中 Y_1 是 Y 的第1列，则 $r = A \leftrightarrow Y = r_1 \leftrightarrow Y'$, Y' 为 Y 中第2列到 k 列组成的 $n \times (k-1)$ 阶矩阵．

5° $\rightarrow$ 是不可换运算，优先级在“+”与矩阵乘之间．

2. 状态方程

设 M 为 Σ 的可达状态，$u_1, u_2, \cdots, u_k$ 为从 M_0 到 M 的变迁步序列，其中每个变迁步都可能是几个变迁的并发，将 $u_1, u_2, \cdots, u_k$ 表示为 m 阶列向量，u_i 的第 j 分量 u_{ij} 为 u_i 中变迁 t_j 的发生次数，$j = 1, 2, \cdots, m$，则(* 为矩阵乘)．

$$M = M_0 \rightarrow C * \begin{bmatrix} u_{11} & u_{21} & \cdots & u_{k1} \\ u_{12} & u_{22} & \cdots & u_{k2} \\ \vdots & \vdots & & \vdots \\ u_{1m} & u_{2m} & \cdots & u_{km} \end{bmatrix}.$$

3. S **不变量**

令 $v_1, v_2, \cdots, v_n$ 为与库所 $s_1, s_2, \cdots, s_n$ 对应的变量，简写 $v = (v_1, v_2, \cdots, v_n)$，则 λ 表达式 $\lambda_v(e)$ 为 Σ 的 S 不变量的条件是：$\lambda_v(e)M = \lambda_v(e)M_0$ 对所有可达标识 M 成立.

例 9　图2-6 中的自控网系统有以下 S 不变量：

$$I_1 = \lambda_v(v_1 + v_2), \qquad I_2 = \lambda_v(v_1 + v_5 = v_2 + v_6),$$
$$I_3 = \lambda_v(v_4 = \text{Fib}(v_6)), \qquad I_4 = \lambda_v(v_3 = \text{Fib}(v_5)),$$

其中 Fib(i) 为 Fibonacci 数列的第 i 项.

I_1 是 P/T 系统意义下的 S 不变量：s_1 和 s_2 中托肯数之和为常量. 由 M_0 知此常量为 1，因此 I_1 的另一形式为 $\lambda_v(v_1 + v_2 = 1)$. I_2、I_3、I_4 中的表达式 e 均为布尔表达式. 在 M_0 这些表达式之值均为 true，所以 I_3、I_4 说明，对任何可达标识 M，$M(s_3) = \text{Fib}(M(s_5))$，$M(s_4) = \text{Fib}(M(s_6))$，$I_2$ 则表明 $M(s_1) + M(s_5) = M(s_2) + M(s_6)$. 所以 S 不变量的另一意义是：令 e 为只含变量 $v_1, v_2, \cdots, v_n$ 的逻辑表达式，则 $\lambda_v(e)$ 为 Σ 的 S 不变量的条件是，对所有可达标识 M，$\lambda_v(e)M$ = true.

4. T **不变量**

$m \times k (k \geqslant 1)$ 阶整矩阵 $\tau = (\tau_{ij})$ 为自控系统 Σ 的 T 不变量的条件是：对所有 $i, j, 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant k, \tau_{ij} \geqslant 0$ 且 $M_0 = M_0 \leftrightarrow C * \tau$.

3　同步论与并发公理

3.1　同步论概念

3.1.1　不同的同步现象

图 3-1 的基本网系统给出了不同的同步现象，(a) 中事件 a 和 b 互为依存，例如同步通信中的收与发，(b) 中事件 a 与 b 交替发生，如走路时的左右腿动作，(c) 中事件 a 和 b 并发，(d) 中事件 a 发生两次 b 发生一次反复循环发生，(e) 中 a 和 b 均可单独发生任意多次.

3.1.2　S 完备化与同步距离

1. S **元素作为观察窗口**

网系统的状态元素又称 S 元素. S 完备化操作使得对任意一对变迁子集 $T_1, T_2 \subseteq T$，只要 $T_1 \cup T_2 \neq \varnothing$，就有一个 S 元素 s 使 $\dot{s} = T_1 \wedge \dot{s} = T_2$. s 可以作为 T_1 事件发生和 T_2 事件发生的同步关系观察窗口. 观察方法是：先在 s 中放上足够多的托肯，在 T_1 中事件发生时 s 中增加一个托肯，T_2 中事件发生时 s 中失去一个托肯，以 s 中托肯数可能的最大值和最小值之差作为对 T_1 和 T_2 中事件发生的同步度量.

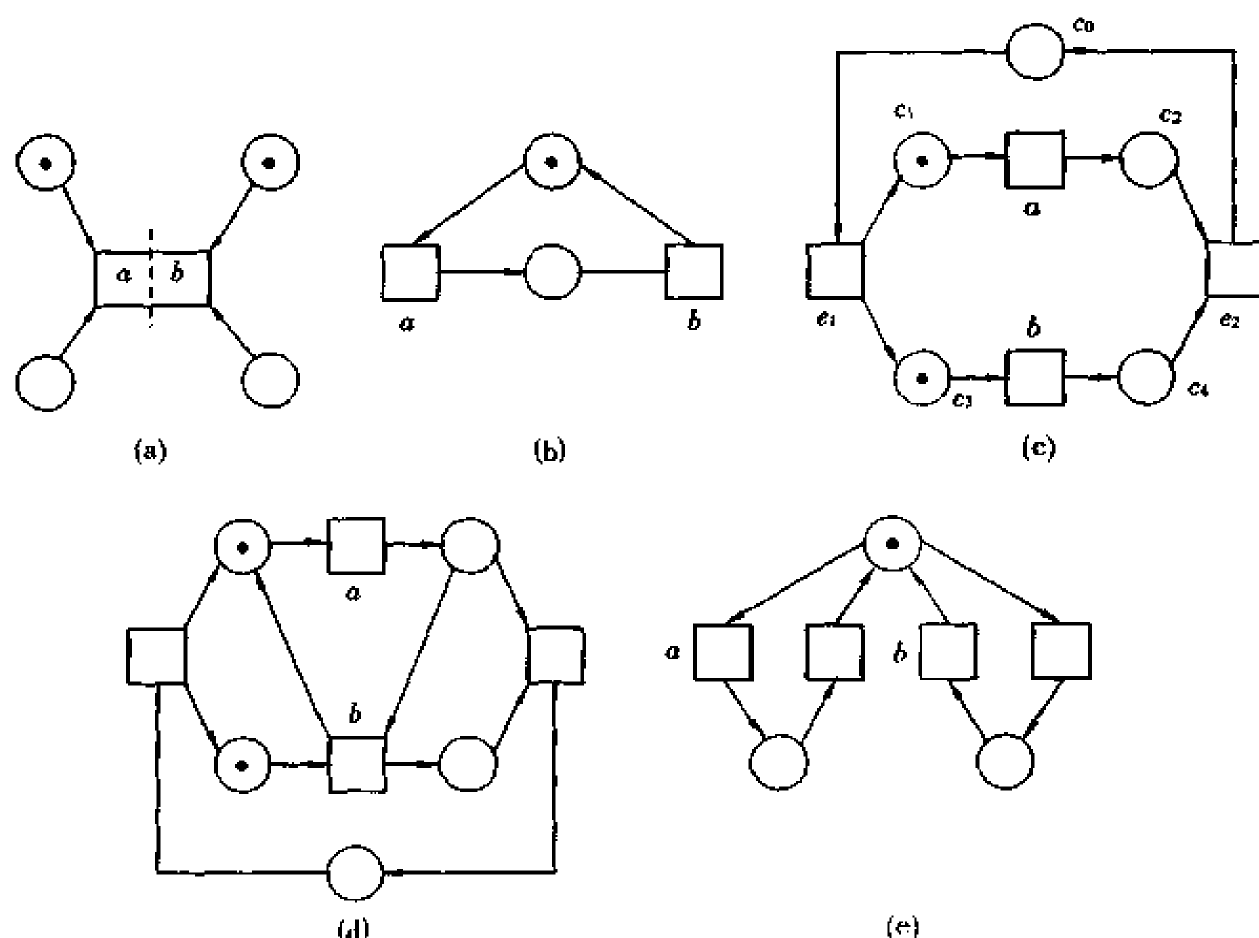


图 3-1

令 $T_1 = \{a\}, T_2 = \{b\}$,则图 3-1 中观察结果是:(a) 中差为 0,(b) 中差为 1,(c) 中差为 2(因为事件 a 发生使 s 得到一个托肯,事件 b 发生使 s 失去一个托肯).(d) 和 (e) 中差为无穷,即不存在最大值.

2. 同步距离概念

图 3-1(a) 中差为 0,表明 a 和 b 不能单独发生,它们是同一事件的两个组成部分;(b) 中差为 1,无论从什么状态开始计数,a 和 b 事件发生次数至多差 1,即 a 和 b 必交替发生;(c) 中从某些状态开始计数(例如图中先让 b 发生以后再计数),一事件可能比另一事件多发生两次(a 可连续发生两次),这就是差为 2 的含义.这个差就是事件 a 和 b 之间的**同步距离**,是对同步现象的定量描述.

3. 事件反向发生

图 3-1(c) 中事件 a 和 b 并发,若从图中删去 e_1、e_2 和 c_0,只保留事件 a、b 及它们的输入输出条件,那么 a 和 b 仍然并发,而其中任何一个都只能发生一次.但 1. 中所述对它们的观察窗口仍会观察到托肯数之差为 2.这是对并发事件相互关系的定量刻画,只与是否并发有关,与是否能第二次发生无关.如果对系统的状态不但考虑其可能的未来,也研究其可能的过去,即让事件反向发生,而观察窗口 s: $\dot{s} = T_1 \wedge \dot{s} = T_2$,则当 T_1 事件正向发生或 T_2 事件反向发生时 s 得到托肯,T_1 事件反向发生或 T_2 事件正向发生时 s 失去托肯,这时 s 中托肯数可能的最大值和最小

值之差就是 T_1 和 T_2 之间的同步距离，记作 $\sigma(T_1,T_2)$.

3.2 条件 / 事件系统

条件 / 事件系统简称 C/E 系统，其特点是没有固定的初始情态. 若把任一情态作为当前情态，则既可以分析它的未来，又可以分析它的过去. 自然界的系统如四季变化都没有初态.

3.2.1 反向发生和挠进程

$(B,E;F)$ 为条件和事件组成的有向网.

(1) 反向发生. 令 c_1、$c_2 \subseteq B$ 为条件丛，$e \in E$ 为事件，若 $c_1[e\rangle c_2$，则说 e 在 c_2 有反向发生权，c_1 是 e 在 c_2 反向发生的反向后继，记作 $c_1\langle e]c_2$.

(2) 反向发生序列. $\sigma = c_1e_1c_2e_2\cdots c_ne_n$ 称为 $(B,E;F)$ 的反向发生序列，如果 $\forall i:1 \leqslant i < n: c_{i+1}\langle e_i]c_i$.

(3) 交错序列. $\sigma = c_1e_1c_2e_2\cdots c_ne_n$ 为交错序列，如果 $\forall i:1 \leqslant i < n: c_i[e_i\rangle c_{i+1} \vee c_{i+1}\langle e_i]c_i$. 显然发生序列和反向发生序列是交错序列的两个极端特例.

(4) 挠进程. 既允许事件正向发生，又允许事件反向发生的进程称为**挠进程**. 挠进程中不能没有事件的反向发生.

(5) 挠进程假设：

1) 若把挠进程中并发的事件任意排定顺序，则得到一个与进程对应的交错序列 σ，交错序列不唯一，但序列中事件的正反向发生次数一样.

2) 若 $\sigma = c_1e_1c_2e_2\cdots c_ne_n$ 是挠进程的一个交错序列，则若有 i、j，$1 \leqslant i < j \leqslant n$，使得 $c_i = c_j$，必有 $j = n$ 或 $\sigma' = c_je_i\cdots e_jc_j$ 中不含事件的反向发生.

3.2.2 完全情态集

设 c_1、$c_2 \subseteq B$ 为条件丛，若有从 c_1 到 c_2 的交错序列，则说 c_1、c_2 之间有**完全可达关系**，记作 $(c_1,c_2) \in R$，R 是 B 的幂集合 $\mathscr{P}(B)$ 上的等价关系.

设 $c \subseteq B$ 为条件丛，以 $[c]$ 表示在完全可达关系 R 下 c 所属的等价类，$[c]$ 称为 $(B,E;F)$ 的一个**完全情态集**. 显然，若 $c' \in [c]$，则 $[c'] = [c]$；若 $c' \in [c]$，则 $[c'] \cap [c] = \varnothing$.

3.2.3 C/E 系统定义

$\Sigma = (B,E;F,C)$ 为 C/E 系统的条件是：

1° $(B,E;F)$ 为简单网，称为 Σ 的基网；

2° C 为 $(B,E;F)$ 的完全情态集，即存在 $c \subseteq B$，使 $C = [c]$. C 中的丛称为情态；

3° $\forall b \in B, \exists c_1、c_2 \in C: b \in c_1 \wedge b \in c_2$；

4° $\forall e \in E, \exists c_1、c_2 \in C: c_1[e\rangle c_2$.

3.2.4　C/E 系统公理

情态公理　$C \neq \varnothing \wedge C \neq \mathscr{P}(B)$，即 C 为 $\mathscr{P}(B)$ 的非空真子集．

外延公理　若 r 是由条件和事件组成的系统中的可达关系：$r \subseteq \mathscr{P}(B) \times E \times \mathscr{P}(B)$，则必有映射 Pre 和 Post：$E \to p(B)$，使得对任何 $e \in E$，记 Pre(e) 为 $^{\cdot}e$，Post$(e) = e^{\cdot}$，必有：

1° $^{\cdot}e \cup e^{\cdot} \neq \varnothing \wedge (^{\cdot}e_1 = {}^{\cdot}e_2 \wedge e_1^{\cdot} = e_2^{\cdot} \Rightarrow e_1 = e_2)$；

2° $(c_1, e, c_2) \in r \Rightarrow c_1 - c_2 = {}^{\cdot}e \wedge c_2 - c_1 = e^{\cdot}$；

3° $^{\cdot}e \subseteq c \wedge e^{\cdot} \cap c = \varnothing \Rightarrow \exists c' : (c, e, c') \in r$；

4° $^{\cdot}e \cap c = \varnothing \wedge e^{\cdot} \subseteq c \Rightarrow \exists c' : (c', e, c) \in r$．

局部确定公理　$\Sigma \cup \mathrm{Env}(\Sigma)$ 作为一个整体是无冲突系统，其中 $\mathrm{Env}(\Sigma)$ 是 Σ 的环境．

3.3　同　步　论

3.3.1　同步距离定义

$\Sigma = (B, E; F, C)$ 为 C/E 系统，E_1、$E_2 \subseteq E$ 且 $E_1 \cup E_2 \neq \varnothing$，则 E_1 和 E_2 之间的同步距离 $\sigma(E_1, E_2)$ 由下式给出：

$$\sigma(E_1, E_2) = \begin{cases} \max\limits_{p \in \pi}\{\,|\,\#(p, E_1) - \#(p, E_2)\,|\,\}, & \text{若最大值存在,} \\ \infty, & \text{若最大值不存在,} \end{cases}$$

其中 π 为 Σ 所有进程的集合，包括挠进程，$\#(p, E_1)$ 和 $\#(p, E_2)$ 分别是 E_1 事件和 E_2 事件在 p 中的发生次数；正向发生为 1，反向发生为 -1．

例 1　图3-2(a) 中 $\sigma(a, b) = 1$，(b) 和(c) 中均为 $\sigma(a, b) = 2$．$\sigma(a, b)$ 是 $\sigma(\{a\}, \{b\})$ 的习惯写法．图 3-1 中(d) 和(e) 都有 $\sigma(a, b) = \infty$．

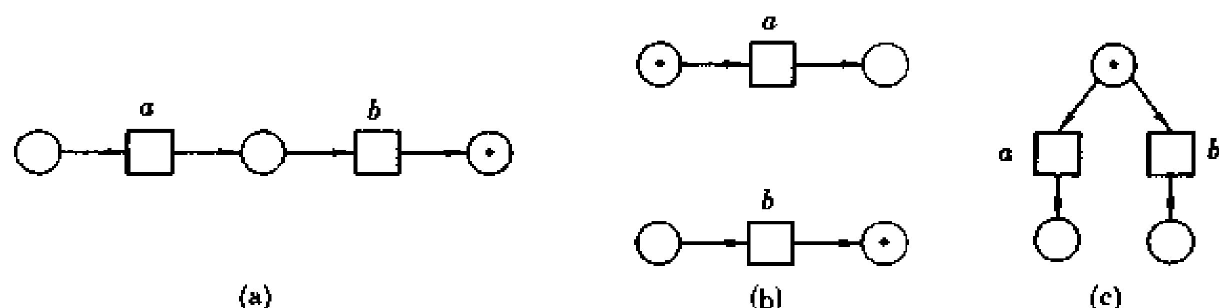


图 3-2

图 3-2 中的托肯给出的是完全情态集中的一个代表情态：等价类由它的任何一个元素唯一确定．

3.3.2　同步距离性质

$\sigma(E_1,E_2)$ 具有包括距离公理在内的以下性质：

(1) $\sigma(E_1,E_2)=\sigma(E_2,E_1)$.

(2) $\sigma(E_1,E_2)=0 \Leftrightarrow E_1=E_2$.

(3) $\sigma(E_1,E_2)\geqslant 0$.

(4) $\sigma(E_1,E_2)\leqslant \sigma(E_1,E_3)+\sigma(E_3,E_2)$.

(5) $\sigma(E_1,E_2)=\sigma(E_1-E_2,E_2-E_1)$.

(6) $\sigma(E_1\cup E_3,E_2\cup E_4)\leqslant \sigma(E_1,E_2)+\sigma(E_3,E_4)+\sigma(E_1\cap E_3,E_2\cap E_4)$.

3.3.3　同步距离计算

1. 完全条件集

若 $\sigma(E_1,E_2)=1$,则令 ${}^{\bullet}b=E_1,b^{\bullet}=E_2$,$b$ 为Σ的一个条件．若 $b\in B$,则 b 为显式给出的条件,若 $b\notin B$,则 b 为隐式给出的条件．显式和隐给出的所有条件之集合 B' 为Σ的完全条件集．对任何 $c\in C$,有唯一的 $c'\in\mathscr{P}(B')$,使 $c\subseteq c'$,且对所有 $b\in B'$,$b\in c'\oplus\bar{b}\in c'$,其中 b' 是 b 的补,即 ${}^{\bullet}b'=b^{\bullet}\wedge b'^{\bullet}={}^{\bullet}b$. c' 称为 $\sum$ 的扩充情态,扩充情态所成之集合记作 C'.

2. 向量表示

为计算 $\sigma(E_1,E_2)$,只须计算 $\sigma(E_1-E_2,E_2-E_1)$. 换言之,可假设 $E_1\cap E_2=\varnothing$. 记 $s=(E_1,E_2)$,即 ${}^{\bullet}s=E_1,s^{\bullet}=E_2$,则 s 与唯一确定的 m 维向量 $\boldsymbol{s}=(u_1,u_2,\cdots,u_m)$ 对应,其中($E=\{e_1,e_2,\cdots,e_m\}$)

$$u_i=\begin{cases}1, & 若\ e_i\in E_1,\\ 0, & 若\ e_i\in E_1\cup E_2,\\ -1, & 若\ e_i\in E_2.\end{cases}$$

$\boldsymbol{s}$ 称为 s 的向量表示．通常不区分 s 和 $\boldsymbol{s}$. 作为 s 的特例,每个条件 $b\in B'$,都有向量表示 $\boldsymbol{b}$.

3. 特征函数

$A:B'\times C'\to\{0,1\}$ 称为$\sum$的特征函数,如果 $\forall b\in B',\forall c'\in C'$,

$$A(b,c')=\begin{cases}1, & 若\ b\in c',\\ 0, & 若\ b\notin c'.\end{cases}$$

4. 计算公式

若 $s=(E_1,E_2)$,s 和 B' 中条件的向量表示 $\boldsymbol{b}$ 线性相关,即存在非负整数 α 和 α_b,使 $\alpha\boldsymbol{s}=\sum\limits_{b\in B'}\alpha_b\boldsymbol{b}$,$\alpha>0$,则有

$$\alpha\sigma(E_1,E_2)=\max_{c'\in C'}\left\{\sum_{b\in B'}\alpha_bA(b,c')\right\}-\min_{c'\in C'}\left\{\sum_{b\in B'}\alpha_bA(b,c')\right\}$$

定理 1　若 $\sigma(E_1,E_2)=\infty$,则 s 和 $b(b\in B')$ 线性无关．

3.3.4 同步距离与系统性质、应用

1. 同步距离与系统性质

同步距离 $\sigma(E_1, E_2)$ 是对 E_1 中事件和 E_2 中事件同步程度的定量描述．

$\sigma(E_1, E_2) = 0$，表明 $E_1 = E_2$，若 E_1 和 E_2 各含一个事件，则这两者在时间和地点上都不可区分．

$\sigma(E_1, E_2) = 1$，E_1 中事件和 E_2 中事件必须交替发生，且 $E_1 \cup E_2$ 中事件无并发．

如果事件 a 和 b 并发，则 $\sigma(a, b) \geqslant 2$.

如果事件 a 和 b 冲突，则 $\sigma(a, b) \geqslant 2$.

2. 应用

(1) 计算同步距离可以分析系统行为．

例 2 图3-3(a) 中信号 A 和 B 不符合同步循环的要求，因为 $\sigma(s_a, s_b) = 2$，信号线 $s_a w_b$ 上有信号重叠的可能. 图 3-3(b) 是通过信号分裂(s_a, s_b) 和等待(w_a, w_b) 实现信号同步的正确方案：$\sigma(s_a, s_b) = 1$.

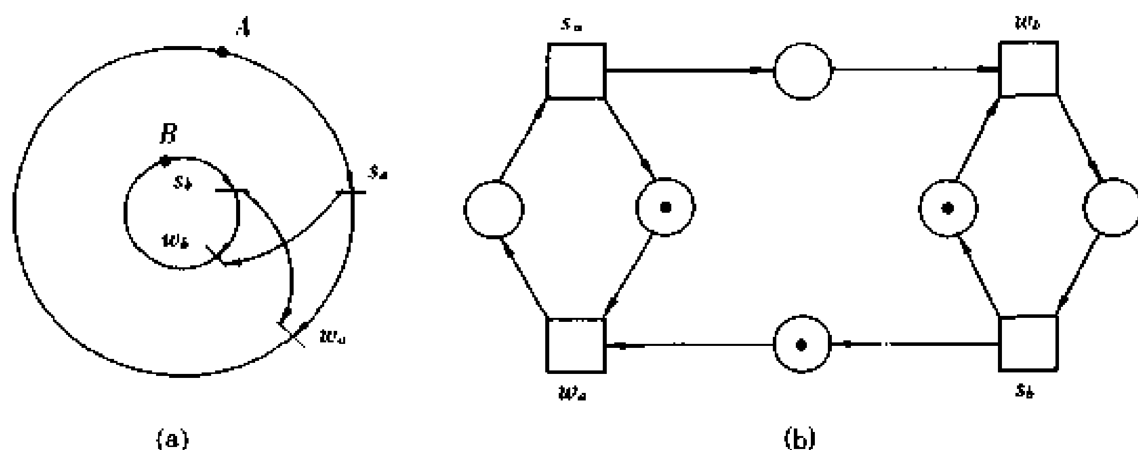


图 3-3

(2) 系统规范．容量为 n 的栈可以用 $\sigma(\text{in}, \text{out}) = n$ 来规范．此规范允许各种实现方案，从先进先出，到若干栈元素的并行进栈(in) 和出栈(out).

3.3.5 加权同步距离

图 3-1(d) 中的 $\sigma(a, b) = \infty$ 和(e) 中的 $\sigma(a, b) = \infty$ 有本质的不同，前者 a 和 b 的发生次数有固定的比例，后者则没有规律．

若存在函数 $\alpha: E \to N$，使 $\max\limits_{p \in \pi} \{ | \#(E_1, \alpha, p) - \#(E_2, \alpha, p) | \}$ 存在

$$\#(E_i, \alpha, p) = \sum_{e \in E_i} \alpha(e) \cdot \#(e, p), \quad (i = 1,2)$$

则
$$\sigma_\alpha(E_1, E_2) = \max_{p \in \pi} \{ | \#(E_1, \alpha, p) - \#(E_2, \alpha, p) | \}$$

称为 E_1 和 E_2 之间的加权同步距离．通常要求 $\{\alpha(e) \mid e \in E\}$ 的最大公约数为 1，以保证 $\sigma_\alpha(E_1, E_2)$ 的唯一性．

例3　图3-1(d)中令$\alpha(a)=1,\alpha(b)=2$,则容易算出$\sigma_\alpha(a,b)=2$.图3-1(e)中的a和b没有这样的函数存在.

加权同步距离是对"零存整取"类行为的定量描述.

3.3.6　同步距离推广

对基本网系统,P/T系统等均可定义同步距离和加权同步距离.这类系统有确定的初始状态,无须考虑挠进程,在定义中只须把进程集π限于正向进程即可.

3.4　并发公理

同步距离对并发事件的定量描述只是并发关系性质的一小部分.

3.4.1　并发概念

(1) 事件并发.组成系统的是事件,例如变迁集T中的变迁和基本网系统的事件集E中的事件.事件会多次发生.并发是事件发生之间的二元关系,不是事件之间的关系.

(2) 资源并发.两个资源个体(或信号),若属于同一系统状态,它们也称为并发(共存的统一称呼).

(3) 资源与事件并发.若资源个体存在于某事件发生之前、之中和之后,称资源个体与该事件并发.

(4) 进程与并发.进程记录的是事件发生与资源个体之间的关系,是对并发关系完整的记录.进程由一个出现网$(B,E;F)$和该网到网系统的映射组成.若把同一事件的不同发生和不同状态下的资源个体当作不同的对象,那么进程描述的是$B\cup E$中元素之间的顺序关系(由F确定)和并发关系(无顺序).换言之,F在$B\cup E$元素之间确定了一个偏序关系.记$X=B\cup E$,用"$<$"表示X上的偏序,并发公理是关于$(X,<)$的公理,$co=\overline{<\cup>}$,其中"$>$"是"$<$"的逆,上横杠为补运算.

3.4.2　并发公理

1. co为类似关系

A0: $|X|\geqslant 2$,非平凡性.

A1: $Id\subseteq co$,自返性,$Id=\{(x,x)\mid x\in X\}$.

A2: $co=co^{-1}$,对称性,$co^{-1}=\{(x,y)\mid(y,x)\in co\}$.

2. 不可化简性

令$li=<\cup Id\cup>=\overline{co}\cup Id$,$li$也是类似关系.

令$Co(x)=\{y\mid(x,y)\in co\}$,定义$\widetilde{co}$如下:$x\,\widetilde{co}\,y:\Leftrightarrow Co(x)=Co(y)$.令$Li(x)=\{y\mid(x,y)\in li\}$,$x\,\widetilde{li}\,y:\Leftrightarrow Li(x)=Li(y)$,

A3: $\widetilde{co}=Id$.

A4: $\widetilde{li} = Id$.

A5: $\widetilde{co} = \overline{li}$（不可化简公理）.

3. 相干性

令 $co^* = Id \cup co \cup co^2 \cup \cdots, li^* = Id \cup li \cup li^2 \cup \cdots$

A6: $co^* = X \times X$.

A7: $li^* = X \times X$.

A8: $co^* = li^*$（相干公理）.

4. 自然序与自然非序

有序关系 $li \subseteq X \times X$ 称为自然序的条件是：任意给定 li 中一对元素的顺序，例如对 $(x,y) \in li$，令 $x < y$，则 li 中所有其它元素 $(u,v) \in li$ 的顺序由 $x < y$ 唯一确定. li 为自然序，则 $co = \overline{li} \cup Id$ 称为自然非序.

A9: (X, co) 是自然非序.

5. 稠密性

X 的切是其元素两两有 co 关系的最大子集，X 的线是其元素两两有 li 关系的最大子集. 每个切和每条线都有交点的性质称为 K 稠密性.

A10: (X, co) 是 K 稠密的.

A11: (X, co) 是 N 稠密的，即

$$\forall x、y、u、v \in X: xliy \wedge y\,li\,u \wedge u\,li\,v \wedge x\,co\,u \wedge x\,co\,v \wedge y\,co\,v$$
$$\Rightarrow \exists z \in X: x\,co\,z \wedge z\,co\,v \wedge y\,li\,z \wedge u\,li\,z.$$

图3-4是 N 稠密性的图示，也是 N 稠密性中"N"的含义：图中 co 和 li 的无向图呈"N"型.

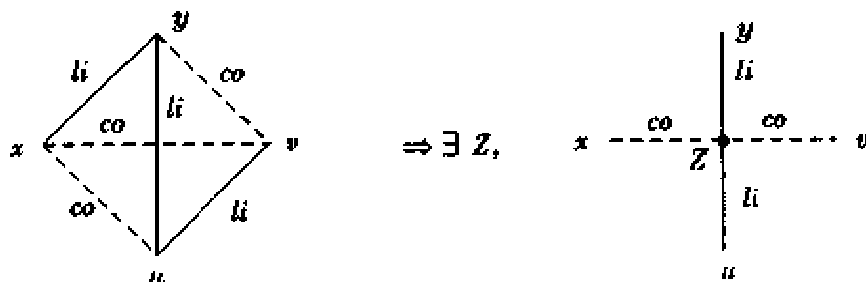


图3-4

6.无向连通网

令 $xCly :\Leftrightarrow Li(x) \subseteq Li(y)$，　$Cl(a): \{y \mid \exists x \in a, xCly\}$，　$P: = Cl - Id, G: = P \cap P^{-1}$.

A12: $P^2 = \varnothing$.

A13: $G^* = X \times X$.

由A12知 $\mathrm{dom}(P) \cap \mathrm{cod}(P) = \varnothing$，$X$ 中点可分为两个不相交的子集 S 和 T. A13说明 (X, P) 是连通的.

7. co 的局部传递性

令 $Vic(x) = \{y \mid xGy\}$，$Vic(x)$ 称为 x 的邻点集，$Vic(x)$ 由 x 所在线上 x 的邻

点组成.

A14: $\forall x \in X$,在 $Vic(x)$ 内 $co^2 \subseteq co$.

A15: $\forall x \in X$,在 $Vic(x)$ 内 $\overline{co}^2 \neq \varnothing \wedge \overline{co}^2 \subseteq co$.

8. 细节

令 $xCl'y :\Leftrightarrow Co(x) \subseteq Co(y)$,

$D := Cl' - Id, H := D \cup D^{-1}$, xDy 表示 x 是 y 的细节.

A16: $D^2 = \varnothing$.

A17: $D^2 = P^2$.

A18: $H^* = X \times X$.

9. 锥形相交

A19: $\forall x, y \in X, x\,\overline{li}\,y \Rightarrow \exists u、v: u < x < v \wedge u < y < v$.

10. 跳与沟:连续性

x 的前锥为小于($<$)等于 x 的点之集合,x 的后锥为大于($>$)或等于 x 的点之集合.若有 $x、y \in X, x < y$,但 x 的前锥加上 y 就是 y 的前锥,y 的后锥加上 x 就是 x 的后锥,就说 x 和 y 之间有个"跳".

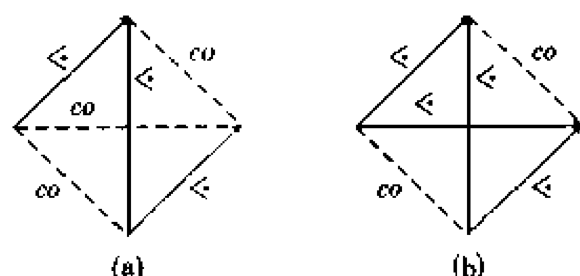
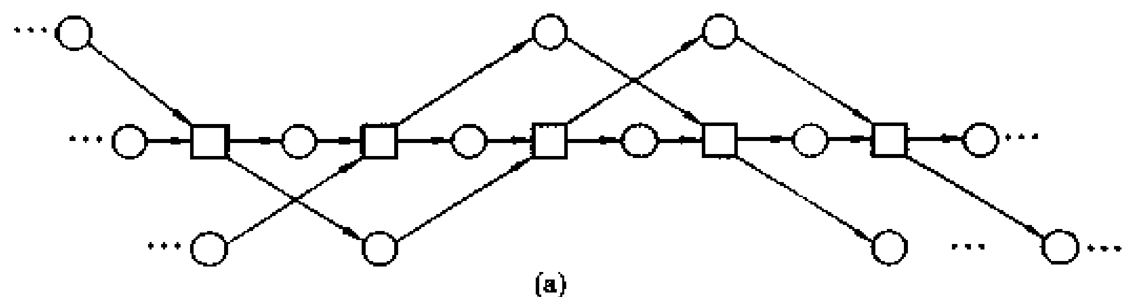


图 3-5

令 $\lessdot = < - <^2$,若$(X, \lessdot)$中存在 N 稠密性中的 N 型结构,但没有相


(a)

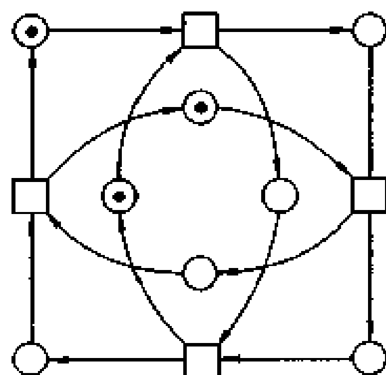

(b)

图 3-6

应的 z 点存在，就说 $(X, <)$ 包含 $G1$ 型沟，如图3-5(a)所示．图3-5(b)为 $G2$ 型沟．

A20：$(X, <)$ 是既无跳又无 $G1$ 型沟的组合顺序．

A21：$(X, <)$ 是既无跳又无 $G2$ 型沟的组合顺序．

所谓组合顺序即为满足 $\leqslant = <^*$ 的严格偏序．

图3-6(a)是满足所有并发公理的最小出现结构，它是图3-6(b)中C/E系统的观察记录．这是描述四季变化的四季系统．

图3-6(a)中结构称为绳(rope)，是独立于观察者的度量"尺"．

4 网 逻 辑

网系统中的变迁决定系统的动态行为．T 完备化操作中构造的变迁形式有些是永无发生权的死变迁，代表着动态系统中的静态性质，例如关于变化中的系统状态的有效命题．

4.1 C/E系统的网逻辑结构

4.1.1 情态和非情态

(1) 条件作为逻辑变量．C/E系统 $\Sigma = (B, E; F, C)$ 的每个条件 $b \in B$ 都既有机会成真，也有机会不成真，因而可作为逻辑变量处理．

(2) 非情态．凡不属于 C 的条件丛都是非情态．非情态集合用 N 表示．

(3) 有效命题．用 B 中条件组成的逻辑表达式，若在所有情态都成真，而在有些非情态不成真，该表达式就是 Σ 的有效命题．

4.1.2 变迁形式分类

1. 变迁形式集

$$T = \{(B_1, B_2) \mid B_1, B_2 \in \mathscr{P}(B) \wedge B_1 \cup B_2 \neq \varnothing\}$$

称为 Σ 的变迁形式集，其中 (B_1, B_2) 表示变迁形式 t，${}^{\cdot}t = B_1$，$t^{\cdot} = B_2$．以下不区分 t 和 (B_1, B_2)．

2. 变迁形式类

粗分为四大类：

$CC = \{t \mid \exists c_1, c_2 \in C : c_1[t\rangle c_2\}$，

$CN = \{t \mid \exists c \in C, \exists u \in N : c[t\rangle u\}$，

$NC = \{t \mid \exists u \in N, \exists c \in C : u[t\rangle c\}$，

$NN = \{t \mid \exists u_1, u_2 \in N : u_1[t\rangle u_2\}$

3. 网逻辑结构 图4-1给出的是 Σ 的网逻辑结构．四大类变迁形式细分为16类．图中大方框代表变迁形式集 T．

4. 进程、事故和事实

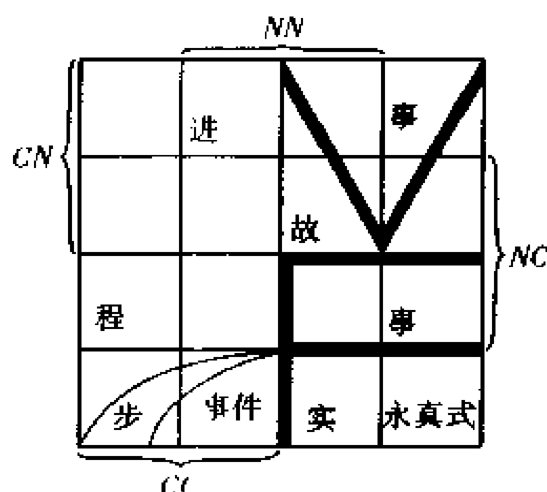


图 4-1

(1) Proc $\stackrel{\text{def}}{=}$ $\{t \mid t \in CC\}$, CC 类的变迁形式代表进程，$^{\cdot}t$ 和 $t^{\cdot}$ 给出的是进程的起止情态. 起止状态相同的进程不属于此类.

(2) Viol: $= \{t \mid t \in CN - CC\}$，这是将情态变为非情态的变迁形式，属事故(violence)类.

(3) Fact: $= \{t \mid t \in \overline{CC \cup CN}\}$，是在任何 $c \in C$ 均不能发生的变迁形式. 代表情态变化中的不变，包括有效命题，称为事实(fact).

(4) Taut: $= \{t \mid t \in \overline{CC \cup CN \cup NC \cup NN}\}$，无论是情态还是非情态都不能使这类变迁形式有发生权，它们代表永真式，是事实类中的特例.

5. 有效命题

$t \in$ Fact 代表关于情态的命题：

若

$$^{\cdot}t = \{a_1, a_2, \cdots, a_n\},$$

$$t^{\cdot} = \{b_1, b_2, \cdots, b_m\}, t \in \text{Fact 等价于}$$

$$a_1 \wedge a_2 \wedge \cdots \wedge a_n \Rightarrow b_1 \vee b_2 \vee \cdots \vee b_m$$

当 $^{\cdot}t \cap t^{\cdot} = \varnothing$ 时，t 代表有效命题. $^{\cdot}t \cap t^{\cdot} \neq \varnothing$ 时，t 为永真式.

4.2 推 理

4.2.1 推理规则

(1) 消解规则. 若 t_1、$t_2 \in$ Fact, $a \in t_1^{\cdot} \cap {}^{\cdot}t_2$，令 $^{\cdot}t_3 = {}^{\cdot}t_1 \cup {}^{\cdot}t_2 - \{a\}$, $t_3^{\cdot} = t_1^{\cdot} \cup t_2^{\cdot} - \{a\}$，则 $t_3 \in$ Fact.

(2) 扩张规则. 若 $t \in$ Fact, u_1、$u_2 \subseteq B$，则 $t' = ({}^{\cdot}t \cup u_1, t^{\cdot} \cup u_2) \in$ Fact.

(3) 矛盾. $t = (\varnothing, \varnothing)$ 代表矛盾.

例 1 对 a、$b \in B$，有效命题 $\neg(a \wedge b)$ 对应的变迁形式是 $t = (\{a, b\}, \varnothing)$. 由扩张规则知 $(\{a, b\}, \{c\}) \in$ Fact，其中 $c \in B$. 这等价于从 $\neg(a \wedge b)$ 推出 $a \wedge b \Rightarrow c$.

例 2 若 $a \wedge b \Rightarrow c \vee d$, $d \Rightarrow x$ 为有效命题，a、b、c、d、$x \in B$，显然 $a \wedge b \Rightarrow c \vee x$ 也是有效命题. 令 $t_1 = (\{a, b\}, \{c, d\})$, $t_2 = (\{d\}, \{x\})$，对 t_1、t_2 用消解规则消去 d，得 $t = (\{a, b\}, \{c, x\})$, $t \in$ Fact. 消解完成了推理.

4.2.2 命题逻辑

定理 1 关于 B 中条件的任何有效命题均可用一个或多个 Fact 类变迁形式表示.

定理 2 反证法. 令 P、Q 为 B 上的命题, 若对 $P \wedge \neg Q$ 的变迁形式表示用消解规则和扩张规则能推出$(\varnothing,\varnothing)$, 即矛盾, 则只要 P 是有效命题, Q 也是.

例 3 例4-2中的推理可用反证法完成: $P \equiv (a \wedge b \Rightarrow c \vee d) \wedge (d \Rightarrow x)$ 等价于 $t_1 = (\{a,b\},\{c,d\})$、$t_2 = (\{d\},\{x\}) \in \text{Fact}$, $Q \equiv a \wedge b \Rightarrow c \vee x$、$\neg Q$ 等价于 $a \wedge b \wedge \neg c \wedge \neg x$, 即 $t_3 = \{\varnothing,\{a\}\}$, $t_4 = (\varnothing,\{b\})$, $t_5 = \{\{c\},\varnothing\}$, $t_6 = \{\{x\},\varnothing\} \in \text{Fact}$. t_2 和 t_6 消解得 $t_8 = (\{d\},\varnothing) \in \text{Fact}$. t_1 和 t_3, t_4, t_5, t_8 消解可得$(\varnothing,\varnothing)$.

定理 3 (1) 若 $t \in \text{Fact}$, 则 t 可用 $m = |B|$ 维列向量表示, 只要 t 不是永真式.

(2) B 上的任何有效命题 P 均可用 $m \times k(k \geqslant 1)$ 阶矩阵表示, k 是 P 对应的变迁形式个数.

(3) 有效命题 P 的矩阵表示可通过矩阵变化得到.

(4) 从 $P \wedge \neg Q$ 推导$(\varnothing,\varnothing)$ 的过程可用矩阵初等变换完成. $(\varnothing,\varnothing)$ 对应着分量全为 0 的列向量.

4.3 P/T 系统和多值逻辑

P/T 系统 $\Sigma = (S,T;F,K,W,M_0)$ 的容量 K 若为常函数, 即 $K \equiv k, k \geqslant 1$ 或 $\forall s \in S: K(s) = k$, 则 Σ 中永无发生权的变迁形式对应着 $k+1$ 值逻辑命题.

4.3.1 变迁形式与多值命题

以 $0,1,\cdots,k$ 表示 $k+1$ 值逻辑的逻辑值, 令变迁形式 $t = (\{s_1,s_2,\cdots,s_n\},\{s_1', s_2',\cdots,s_m'\})$, 若 t 在任何可达标识 $M \in [M_0\rangle$ 中均无发生权, 则它等价于命题.

$$\neg(\forall i: 1 \leqslant i \leqslant n: a_i \geqslant v_i) \vee \neg(\forall j: 1 \leqslant j \leqslant m: b_j + w_j \leqslant k),$$

其中 $a_i = M(s_i)$, $b_j = M(s_j')$, $v_i = W(s_i,t)$, $w_j = W(t,s_j')$, $i = 1,2,\cdots,n$, $j = 1,2,\cdots,m$.

例 4 令 $K \equiv 2$, Σ 对应三值逻辑. 若 $t_1 = (\{s_1\},\varnothing\}$, $t_2 = (\{s_1\},\{s_2\})$, $t_3 = (\{s_2\},\{s_1\})$, 且 $W(s_1,t_1) = W(t_3,s_1) = 2$, $W(s_1,t_2) = W(t_2,s_2) = W(s_2,t_3) = 1$, $M(s_1) = a$, $M(s_2) = b$, 则 t_1、t_2 和 t_3 永无发生权的事实等价于

$$t_1: a < 2, t_2: a < 1 \vee b > 1, t_3: a > 0 \vee b < 1.$$

由 $K \equiv 2$ 知 $0 \leqslant a$、$b \leqslant 2$, 从而上述不等式等价于 $2a = b$ 或 $2a - b = 0$. 这是三值逻辑系统的命题, 当然, 对于三值逻辑运算(0、1 和 2 上的算术运算) 必须有合理的解释.

把 S 中的状态元素看作整型变量, 其值不超过 k, 则 Σ 的永无发生权的变迁形

式是关于这些变量依赖关系的命题.

5 信息流结构

基本网系统可以作成可逆的信息元件.

5.1 基本信息流图

5.1.1 箭头函数

1. 第一箭头函数 P_1

令 a、b 为一位信息,即 a、b 的值为 1 或 0. P_1 以 a 和 b 为输入,其功能是使 a 作用于 b,使其改变为信息 $u = b \oplus a$. 信息 a 作为另一输出不受影响.

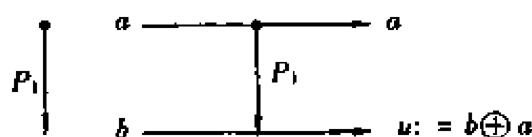


图 5-1

图 5-1 是 P_1 的信息流图. 箭头所指为受影响的信息,黑点则是施加影响的信息.

2. 第 n 箭头函数 P_n

信息 $x_1, x_2, \cdots, x_n (n > 1)$ 及 y 为输入, $x_1, x_2, \cdots, x_n$ 和 u 为输出,

$$u = y \oplus \prod_{i=1}^{n} x_i,$$

其中 $\prod_{i=1}^{n} x_i$ 可按算术乘法运算.

图 5-2(a) 是 P_n 的信息流图. 图 5-2(b) 为 P_2.

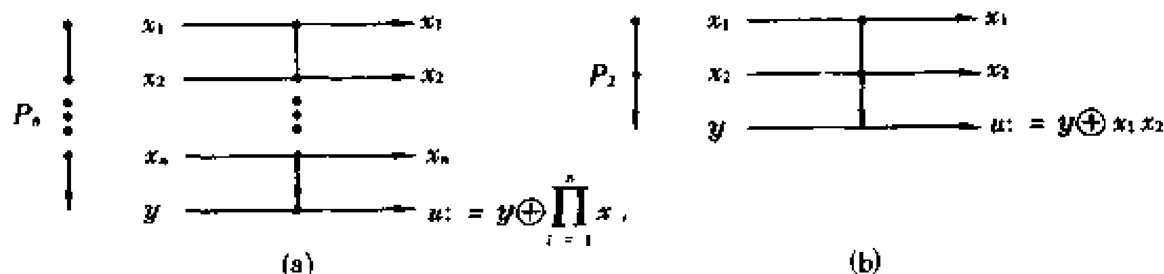


图 5-2

3. 条件交换函数 Q

Q 以 a、b、c 为输入,信息 b 同时影响 a 和 c:在 $b = 1$ 时交换 a 和 c.

图 5-3 是 Q 的信息流图和定义. 其中 $\bar{b} = 1 - b$.

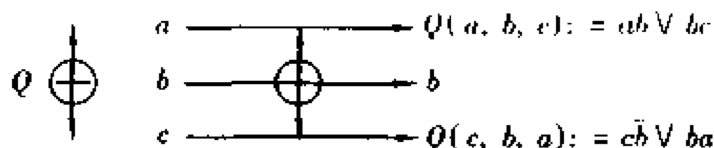


图 5-3

4. P_0

P_n 的定义可以推广到 $n = 0$，只要注意到 $\prod_{i=1}^{0} x_i = 1$，于是 P_0 的输入 y 转换为

$$y \oplus \prod_{i=1}^{0} x_i = \bar{y}.$$

定理 1　(1) $P_n(n \geqslant 0)$ 和 Q 都是可逆的：

$P_n^2 = Id \wedge Q^2 = Id$ 即 $P_n^{-1} = P_n \wedge Q^{-1} = Q$，其中 Id 为无信息交互作用的流动.

(2) 当 $n > 2$ 时，一切 P_n 均可用 P_2 构造出来.

(3) 一切逻辑函数 $f:2^n \to 2^m$ 均可用箭头函数构造或表示.

(4) 用 P_1 和 P_2 可以构造 Q，用 P_1 和 Q 可以构造 P_2，所以 P_1 和 Q 可以作为信息元件.

5.1.2　常信息和信息站

(1) 常信息. 信息函数 P_n 和 Q 中的输入 a、b、x、y 等都是信息变量. 常信息有两种，一是信息值不变的常量信息，一是信息值不参与交互作用的信息.

(2) 信息站. 基本网系统的条件只有两种状态，但信息有三种状态：首先是有无信息，有信息时又分 1 和 0 两个值. 信息的表示如图 5-4 所示，称为信息站. 图中的 ⊢ 表示永无发生权的变迁形式，所以图(a) 中不允许 $a = 0$ 和 $a = 1$ 同时成真，图(b) 中是常信息站，与 ⊢ 有实箭头相连的 $a = 0$ 永不会成真：表示常信息 $a = 1$ 或无信息.

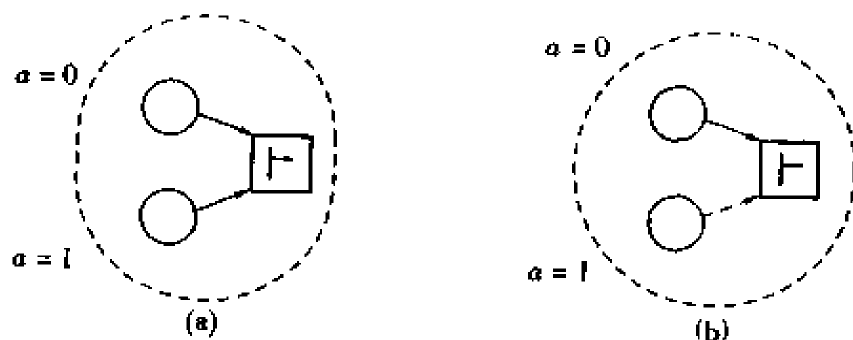


图 5-4

5.2　信息流图的有向网表示

5.2.1　P_1 的网表示

把 P_1 看成变迁，得图 5-5，细化得图 5-6.

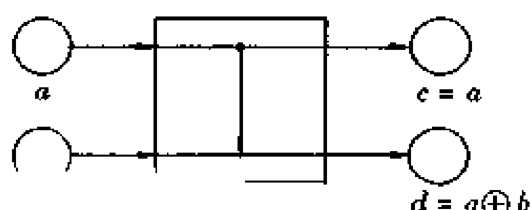


图 5-5

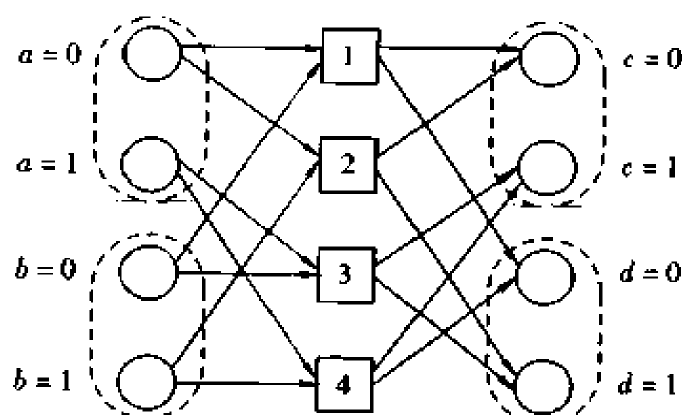


图 5-6

5.2.2　Q 的网表示

图 5-7 是 Q 的网表示. 图中 e' 和 b' 为 Q 内部信息，而且是同一个信息. a、b、c 是输入信息，d、e、f 是输出信息，$d = Q(a,b,c)$，$f = Q(c,b,a)$，$e = b$.

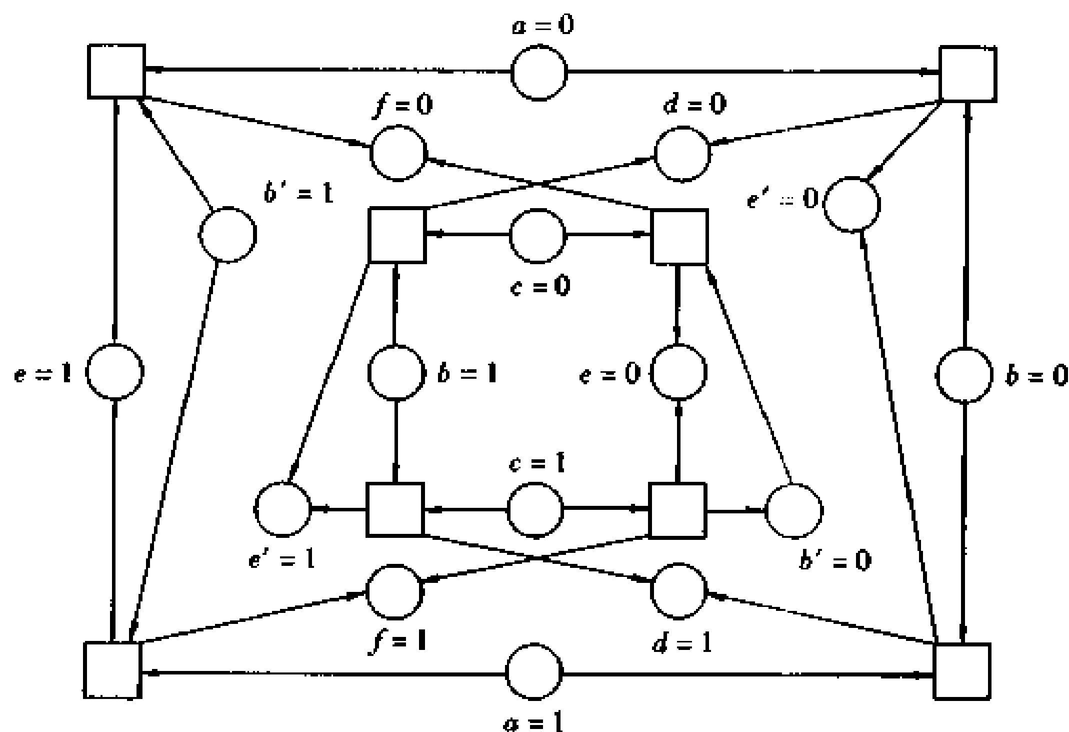


图 5-7

5.3 信息元件:一位噪音通道

5.3.1 一位噪音通道定义

图5-8给出的是两个一位噪音通道的网表示.由于两者输出与输入之间没有固定的联系,所以称为噪音通道.

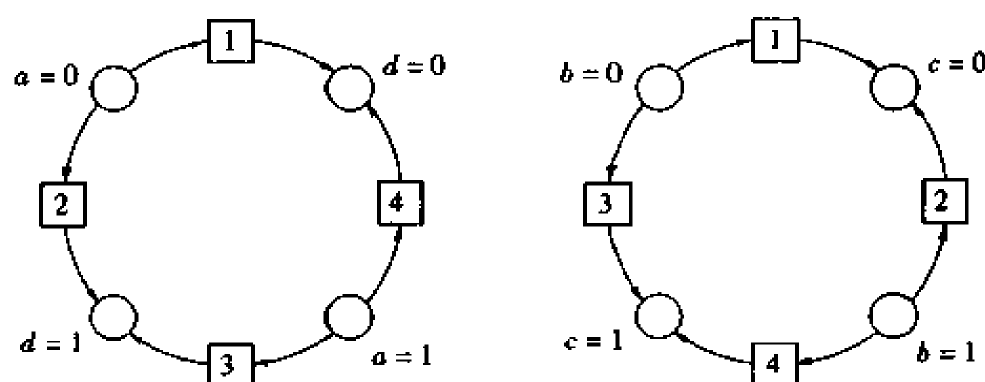


图 5-8

5.3.2 信息元件

定理2 (1) P_1 由两个一位噪音通道合成.

(2) Q 由四个一位噪音通道合成.

(3) 一切逻辑函数 $f:2^n \to 2^m$ 均可由一位噪音通道表示.

(4) 只用一位噪音通道即可以构造出可逆信息系统的唯一信息元件.

参考文献

1 Petri C A. Communication with automata. Final Report, Volume 1, supplement 1. Applied Data Reséarch, Englewood Cliffs: Princeton, 1966.

2 Petri C A. Concepts of net theory. Mathematical Foundations of Computen Science, 1973: 137 ~ 146

3 Petri C A. General net theory. Proceedings of the Joint IBM Univesity of Newcastle Upon Tyne Seminar on Computs System Design, 1977.

4 Peterson J L. Petri net theory and the modeling of systems. Englewood Cliffs: Prentice-Hall, Inc, 1981.

5 Reisig W. Petri nets. Beilin: Springer-Verlag, 1985.

6 袁崇义著. Petri网. 南京:东南大学出版社,1989.

7 袁崇义著. Petri网原理. 北京:电子工业出版社,1998.

·计算机数学卷·

第9篇

网络最优化

编　者　刘振宏
审校者　卢开澄

目　录

引　言

网络分析是研究如何用网络的方法刻划和求解最优化问题，特别是离散最优化问题．由于网络方法直观易懂和便于操作，所以网络方法越来越受到人们的重视．目前它已成为运筹学、计算机科学、管理科学和系统工程中不可缺少的一部分．

网络分析一般认为是起源于网络流的研究．网络流理论作为一门科学分支出现并引起人们的重视，应该归功于福特（Ford）和冯克孙（Fulkerson）的名著 Flows in Networks，该书从线性规划的理论出发，系统地讨论了若干纯组合的数学问题和网络流的若干问题．它不仅是网络分析发展的根基，同时也是整数线性规划发展的一个里程碑．我国在 20 世纪 50 年代末期和 60 年代初期也做出了一些在国际上有影响的工作．例如，物资调运的图上作业法，中国邮递员问题以及有向网络中的最小树形图问题和树网络的重心问题等．正是在这些工作的基础上，开展了我国图论的研究与发展．

本篇只叙述网络分析中一些重要问题的算法，而算法的证明不做重点讨论，感兴趣的读者可查阅有关文献．

0　图的几种表示形式

在用计算机求解网络问题时，总是要把图输入计算机，因此图的表示方式与计算效率有直接关系．图的表示有多种形式，根据问题的类型不同，可以选择恰当的表示形式，以提高其计算效率．

设 $G(V,E)$ 和 $D(V,A)$ 分别是无向图和有向图，$V=\{1,2,\cdots,n\}$ 为顶点集，$E=\{e_1,e_2,\cdots,e_m\}$ 为边集，$A=\{a_1,a_2,\cdots,a_m\}$ 为弧集．那么 G 和 D 通常有下述几种表示方法：

1. 邻接矩阵

设 $\boldsymbol{M}(G)$ 和 $\boldsymbol{M}(D)$ 是两个 $n\times n$ 的方阵，它们的行和列都对应图的顶点．记 $\boldsymbol{M}(G)=[m_{ij}]$，$\boldsymbol{M}(D)=[n_{ij}]$，定义

$$m_{ij}=\begin{cases}1, & 若[i,j]\in E,\\ 0, & 其余,\end{cases}$$

$$n_{ij}=\begin{cases}1, & 若(i,j)\in A,\\ 0, & 其余.\end{cases}$$

则称 $\boldsymbol{M}(G)$ 和 $\boldsymbol{M}(D)$ 分别为 G 和 D 的**邻接矩阵**．这儿用 $[i,j]$ 表示端点为 i 和 j 的边；而 (i,j) 表示自 i 到 j 的弧．值得注意的是，边 $[i,j]$ 是无序对，而弧 (i,j) 是有序对，即 $[i,j]=[j,i]$，但 $(i,j)\neq(j,i)$．

例如图 0-1 和图 0-2 的邻接矩阵如下：

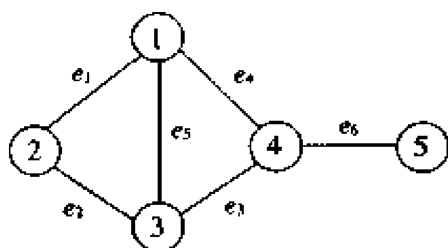


图 0-1

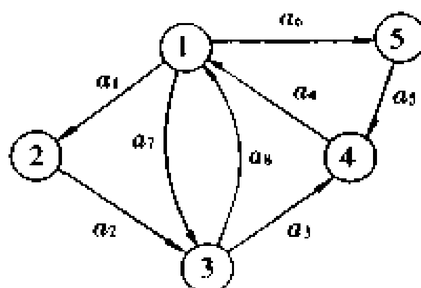


图 0-2

$$\boldsymbol{M}(G)=\begin{bmatrix}0&1&1&1&0\\1&0&1&0&0\\1&1&0&1&0\\1&0&1&0&1\\0&0&0&1&0\end{bmatrix},\qquad \boldsymbol{M}(D)=\begin{bmatrix}0&1&1&0&1\\0&0&1&0&0\\1&0&0&1&0\\1&0&0&0&0\\0&0&0&1&0\end{bmatrix}.$$

当 G 是多重图(即两个顶点间有多条边)时,则 G 的邻接矩阵 $\boldsymbol{M}(G)=[m_{ij}]$ 的定义为:m_{ij} 等于 i 和 j 之间的边的条数;类似地,若 D 为多重有向图时,n_{ij} 等于顶点 i 到 j 的弧的条数.由此可见,不管 G 是简单图还是多重图,$\boldsymbol{M}(G)$ 总是对称的.当 G 或 D 是简单图或有向图时,则 $\boldsymbol{M}(G)$ 和 $\boldsymbol{M}(D)$ 都是 0-1 矩阵.

2. 关联矩阵

设 $G(D)$ 是 n 个顶点 m 条边(弧)的图(有向图),$\boldsymbol{B}(G)=[m_{ij}]$ 和 $\boldsymbol{B}(D)=[n_{ij}]$ 是两个 $n\times m$ 阶矩阵,其定义为

$$m_{ij}=\begin{cases}1, & \text{若顶点 } i \text{ 是边 } e_j \text{ 的端点},\\0, & \text{其余},\end{cases}$$

$$n_{ij}=\begin{cases}1, & \text{若 } i \text{ 是弧 } a_j \text{ 的始点},\\-1, & \text{若 } i \text{ 是弧 } a_j \text{ 的终点},\\0, & \text{其余}.\end{cases}$$

则称 $\boldsymbol{B}(G)$ 和 $\boldsymbol{B}(D)$ 分别为图 G 和 D 的**关联矩阵**.

例如图 0-1 和 0-2 的关联矩阵分别为

$$\boldsymbol{B}(G)=\begin{bmatrix}1&0&0&1&1&0\\1&1&0&0&0&0\\0&1&1&0&1&0\\0&0&1&1&0&1\\0&0&0&0&0&1\end{bmatrix},\ \boldsymbol{B}(D)=\begin{bmatrix}1&0&0&-1&0&1&1&-1\\-1&1&0&0&0&0&0&0\\0&-1&1&0&0&0&-1&1\\0&0&-1&1&-1&0&0&0\\0&0&0&0&1&-1&0&0\end{bmatrix}.$$

有向图 D 的关联矩阵 $\boldsymbol{B}(D)$ 具有**全单位模性质**,即 $\boldsymbol{B}(D)$ 的每一个非奇异的子方阵的行列式之值为 ± 1.这一性质保证了网络中若干线性规划问题有整数解.同样,当 G 为二部图时,关联矩阵 $\boldsymbol{B}(G)$ 也是全单位模的,因此运输问题有整数解.

图的邻接矩阵和关联矩阵是图的最常用的两种表示方法,此外还有

3. 边目录与弧目录

边(弧)目录是指按图的顶点编号的字典序列出其所有边(弧).如图 0-1 和 0-2

的边目录和弧目录分别为

$$[1,2][1,3][1,4];[2,3];[3,4];[4,5]$$

和

$$(1,2)(1,3)(1,5);(2,3);(3,1)(3,4);(4,1);(5,4).$$

边目录和弧目录也可以用两行阵的形式来表示.例如图0-1和0-2的两行阵表示分别为

$$\begin{bmatrix} 1 & 1 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 3 & 4 & 5 \end{bmatrix}$$

及

$$\begin{bmatrix} 1 & 1 & 1 & 2 & 3 & 3 & 4 & 5 \\ 2 & 3 & 5 & 3 & 1 & 4 & 1 & 4 \end{bmatrix}.$$

上述各种表示形式各有其特点,至于哪种形式好,要视具体问题而定.

1 最短路问题

1.1 最短路问题的数学模型

设 $N(V,A;w)$ 为一个有向网络,$V=\{1,2,\cdots,n\}$ 为顶点集,A 为弧集.w 为弧集 A 上的权函数,即对每一弧 $(i,j)\in A$,有一个权 w_{ij},它可以是弧 (i,j) 的长度或通过 (i,j) 的时间或费用等.从一个顶点 i 到另一个顶点 j 的**路的长度**定义为该路上弧的权之和.从顶点 i 到顶点 j 长度最短的路称为 i 到 j 的**最短路**.**最短路问题**就是在一个网络中求一个指定点到另一个指定点的最短路.最短路问题不仅其自身有重要应用,而且许多组合优化问题的算法以最短路算法做为其子程序.

网络 $N(V,A;w)$ 中自顶点 1 到顶点 n 的最短路问题,可描述为下述模型:

$$\begin{cases} \min \sum\limits_{(i,j)\in A} w_{ij}x_{ij}, \\ \sum\limits_{j\in\alpha(i)} x_{ij} - \sum\limits_{j\in\beta(i)} x_{ji} = \begin{cases} 1, & i=1, \\ 0, & i\neq 1,n, \\ -1, & i=n, \end{cases} \\ x_{ij}\geqslant 0, \quad (i,j)\in A, \end{cases} \tag{1-1}$$

其中 $\alpha(i)=\{j\in V\mid (i,j)\in A\}$ 称为顶点 i 的**外邻点集**,而 $\beta(i)=\{j\in V\mid (j,i)\in A\}$ 称为顶点 i 的**内邻点集**.

易证,模型(1-1)有可行解,当且仅当网络 N 中存在自顶点 1 到顶点 n 的路.进而,模型(1-1)有有界最优解,当且仅当 N 中有自顶点 1 到顶点 n 的路,并且 N 中不含**负回路**(权和为负的有向圈).

另外,由于模型(1-1)中约束条件的系数矩阵是有向图 $D(V,A)$ 的关联矩阵 $\boldsymbol{B}(D)$.$\boldsymbol{B}(D)$ 的单位模性质保证了模型(1-1)有整数最优解.事实上它有 0-1 最优解.假设 $\{x_{ij}\}$ 为模型(1-1)的最优解,则弧集

$$\{(i,j)\mid x_{ij}^0=1\}$$

组成网络中自顶点 1 到顶点 n 的最短路. 因此最短路问题可以用线性规划的单纯形算法或内点法求解. 然而从线性规划的原始 - 对偶算法的思想出发, 结合网络的特点, 可以得到简单地纯组合算法. 为此考察模型(1-1) 的对偶问题:

$$\begin{cases}\max\{u_n - u_1\}, \\ \quad u_j - u_i \leqslant w_{ij}, \quad (i,j) \in A.\end{cases} \tag{1-2}$$

由于模型(1-1) 的系数矩阵非满秩, 故(1-2) 式中有多余的变量, 不妨设 $u_1 = 0$. 这样(1-2) 式可写为

$$\begin{cases}\max\ u_n, \\ \quad u_j - u_i \leqslant w_{ij}, \\ \quad u_1 = 0.\end{cases} \tag{1-3}$$

此时 u_j 可以理解为自顶点 1 到顶点 j 的最短路长度. 显然, 若 u_j 表示自顶点 1 到顶点 j 的最短路长度, $j = 2,3,\cdots,n$, 那么 $\{u_j\}$ 是(1-3) 式的可行解. 但反之不对. 然而, 当 N 中自点 1 到每一点 j 有路, 并且 N 中无负回路时, u_j 是顶点 1 到顶点 j 的最短路长度, $j = 2,3,\cdots,n$, 当且仅当它们满足

$$\begin{cases}u_j = \min\limits_{1\leqslant k\leqslant n}\{u_k + w_{kj}\}, \quad j = 2,3,\cdots,n, \\ u_1 = 0.\end{cases} \tag{1-4}$$

注意这儿规定 $w_{ii} = 0, i = 1,2,\cdots,n$. 由于方程组(1-4) 是非线性隐函数的, 所以直接求解很困难, 一般均采用逐次逼近法求解. 但对某些特殊情况有一些简单的算法.

1.2 非负权网络中最短路算法

设网络 $N(V,A;w)$ 中每一弧 $(i,j) \in A$ 的权 $w_{ij} \geqslant 0$, 若 $(i,j) \notin A$, 则令 $w_{ij} = +\infty$. 1959 年迪克斯特拉(E. Djekstra) 对 $w_{ij} \geqslant 0$ 的情况给出了逐点确定最短路的算法如下:

步 0　置 $S \leftarrow \{1\}, T = \{2,3,\cdots,n\}, \boldsymbol{R} = (J_1, J_2, \cdots, J_n)$; 对一切 $i = 1,2,\cdots,n, J_i = 1$; $u_1 = 0, u_j = w_{1j}, j \in T$.

步 1　在 T 中找一点 k, 使 $u_k = \min\limits_{j\in T} u_j$; 置 $S \leftarrow S \cup \{k\}, T \leftarrow T - \{k\}$.

若 $T = \varnothing$, 转步 3. 其中 $\varnothing$ 表示空集.

步 2　对一切 $j \in T$ 修正 u_j 如下:

若 $u_j > u_k + w_{kj}$, 置 $u_j \leftarrow u_k + w_{kj}, J_j \leftarrow k$;

若 $u_j \leqslant u_k + w_{kj}$, 则 u_j 不变.

返回步 1.

步 3　找出自顶点 1 到其余各点的最短路.

注意, 对一切 $j = 2,3,\cdots,n$, $\boldsymbol{R}$ 中第 j 个分量 J_j 是自顶点 1 到顶点 j 的最短路上倒数第 2 个顶点. 由 J_j 的意义不难找出顶点 1 到任一顶点 j 的最短路.

例 1　求图1-1中自顶点 1 到各点的最短路.

步 0　置 $S = \{1\}, T = \{2,3,4,5,6,7\}, \boldsymbol{R} = (1,1,1,1,1,1,1)$; $u_1 = 0, u_2 = 4$,

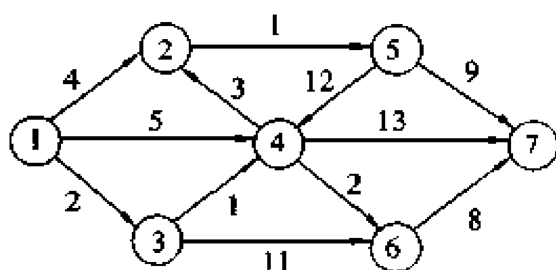


图 1-1

$u_3 = 2, u_4 = 5, u_5 = u_6 = u_7 = +\infty$.

步 1 $\min\limits_{j \in T}\{u_j\} = u_3$,即 $k = 3$;此时 $S = \{1,3\}, T = \{2,4,5,6,7\}$.

步 2 由于 $u_4 = 5 > u_3 + w_{34} = 3, u_6 = \infty > u_3 + w_{36} = 13$,故得新的 $u_1 = 0, u_2 = 4, u_3 = 2, u_4 = 3, u_6 = 13, u_5 = u_7 = +\infty$;$\boldsymbol{R} = (1,1,1,3,1,3,1)$.然后使用新的$\{u_j\}$、$S$、$T$、$\boldsymbol{R}$ 返回步 1.

经过 6 次迭代后,得 $T = \varnothing, \boldsymbol{R} = (1,1,1,3,2,4,6)$;$u_1 = 0, u_2 = 4, u_3 = 2, u_4 = 3, u_5 = 5, u_6 = 5, u_7 = 13$.此时诸 u_j 就等于顶点 1 到顶点 j 的最短路长度.

步 3 根据 $\boldsymbol{R}$ 求出自顶点 1 到其余各点的最短路.如 1 到 7 的最短路:$J_7 = 6, J_6 = 4, J_4 = 3, J_3 = 1$,故这条路为 $1 \to 3 \to 4 \to 6 \to 7$.其长为 $u_7 = 13$.

不难看出迪克斯特拉算法的计算复杂性为 $O(n^2)$.

1.3 无回路网络中最短路算法

任何一个无回路有向网络 $N(V,A;w)$,总可以对其顶点进行编号,使得对任意 i、$j \in V$,若 $i < j$,则$(j,i) \notin A$.因为无回路的网络总存在入次为0的顶点.首先对入次为 0 的顶点给标号 1,然后去掉标号的顶点及关联的弧,对剩下的网络中入次为 0 的点给标号 2,3,…,重复这个过程,直到所有点给予标号为止.

如果一个无回路网络的顶点标号满足:若 $i < j$,则$(j,i) \notin A$,那么自顶点 1 到顶点 j 的最短路必包含在由顶点集$\{1,2,\cdots,j\}$ 构成的子网络里.因此也可逐点求出其最短路.

设 $N(V,A;w)$ 是无回路有向网络,设自顶点 1 到每一个顶点都有路.求自顶点 1 到各顶点的最短路算法如下:

步 0 置 $S = \{1\}, T = \{2,3,\cdots,n\}, \boldsymbol{R} = (J_1, J_2, \cdots, J_n)$;对一切 $i = 1,2,\cdots, n, J_i = 1$;$u_1 = 0, u_j = w_{1j}, j \in T$;$\boldsymbol{M}(D) = [n_{ij}]$为邻接矩阵;$V_T = \{\sum\limits_{i=2}^{n} n_{ij} = d_j \mid j \in T\}$.应注意,当$(i,j) \notin A$ 时,定义 $w_{ij} = +\infty$.

步 1 在 T 中取一点 k,使 $d_k = 0$.置 $S \leftarrow S \cup \{k\}, T \leftarrow T - \{k\}, V_T \leftarrow V_T - \{d_k\}$;若 $T = \varnothing$,转第 3 步.

步 2 (修正 u_j 和 d_j) 对每一个 $j \in T$,修正 u_j 和 d_j 如下:

若 $u_j > u_k + w_{kj}$，置 $u_j \leftarrow u_k + w_{kj}$，$J_j \leftarrow k$；
若 $u_j \leqslant u_k + w_{kj}$，则 u_j 和 J_i 不变；
对一切 $j \in T$，置 $d_j \leftarrow d_j - n_{kj}$，返回第1步.
步3　由 R 找出自顶点1到各点的最短路.
例2　求图1-2中自顶点1到各点最短路.

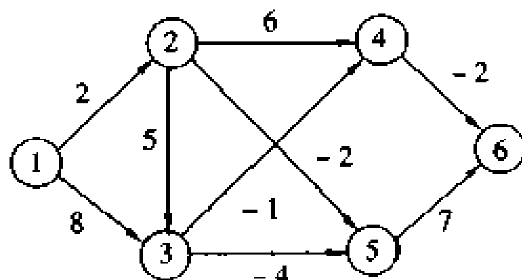


图 1-2

步0　$S = \{1\}$，$T = \{2,3,4,5,6\}$，$R = (J_1, J_2, \cdots, J_6) = (1,1,\cdots,1)$；$u_1 = 0$，$u_2 = 2$，$u_3 = 8$，$u_4 = u_5 = u_6 = +\infty$；而 $V_T = \{d_2, d_3, d_4, d_5, d_4\} = \{0,1,2,2,2\}$.

步1　由于 $d_2 = 0$，即 $k = 2$，置 $S = \{1,2\}$，而 $T = \{3.4,5,6\}$；$V_T = \{d_3, d_4, d_5, d_6\}$. 由于 $T \neq \varnothing$，进行步2.

步2　因 $u_3 < u_2 + w_{23} = 7$，$u_4 < u_2 + w_{24} = 8$，$u_5 < u_2 + w_{25} = 0$，从而置 $u_3 = 7$，$u_4 = 8$，$u_5 = 0$；其余 $u_1 = 0$，$u_2 = 2$，$u_6 = +\infty$ 不变；而 $J_3 \leftarrow 2$，$J_4 \leftarrow 2$，$J_5 \leftarrow 2$；$d_3 \leftarrow d_3 - n_{23} = 0$，$d_4 \leftarrow d_4 - n_{24} = 1$，$d_5 \leftarrow d_5 - n_{25} = 1$. $d_6 \leftarrow d_6 - n_{26} = 2$，即 $R = (1,1,2,2,2,1)$，$V_7 = \{0,1,1,2\}$. 然后返回步1.

经过5次迭代后 $T = \varnothing$；此时 $R = (1,1,2,3,2,4)$；$u_2 = 2$，$u_3 = 7$，$u_4 = 6$，$u_5 = 0$，$u_6 = 4$.

步3　根据记录 R 找出顶点1到各顶点的最短路. 例如顶点1到顶点6的最短路为

$$① \longrightarrow ② \longrightarrow ③ \longrightarrow ④ \longrightarrow ⑥$$

由于网络中无回路，故弧的权无论正负，总是不存在负回路的，从而对无回路网络，上述算法也可求出两点间的最长路. 这只要把权 w_{ij} 变为 $-w_{ij}$，那么求出的最短路就是原网络中的最长路.

不难看出，无回路网络的最短路算法其计算复杂性也是 $O(n^2)$.

1.4　无负回路网络中最短路算法

当网络中的弧允许有负权，但没有负回路时，1.2节中的迪克斯特拉算法就会失效，因此必须建立新的算法.

设 u_{1j}^l 表示网络中自顶点1到顶点 j 最多由 l 条弧组成的一切路中最短一条路的长度. 因此对一切 $j = 2,3,\cdots,n$，必有

$$u_{1j}^1 \geqslant u_{1j}^2 \geqslant \cdots \geqslant u_{1j}^{n-1}.$$

而且由于网络中无负回路，那么对任意 $l \geqslant n-1$，必有 $u_{1j}^l = u_{1j}^{n-1}$. 因此 u_{1j}^{n-1} 必定

是网络中自顶点 1 到顶点 j 的最短路长度. 具体算法如下:

步 0 置 $l \leftarrow 1, u_{11}^l = 0, u_{1j}^l = w_{1j}, j = 2,3,\cdots,n$(这儿定义 $w_{ii} = 0, i = 1,2,\cdots,n$, 若 $(i,j) \notin A, w_{ij} = +\infty$), $\boldsymbol{R} = (J_1, J_2, \cdots, J_n)$, 而 $J_i = 1, i = 1,2,\cdots,n$.

步 1 对每一个顶点 $j = 2,3,\cdots,n$, 找出一个顶点 k, 使得

$$u_{1k}^l + w_{kj} = \min_{1 \le i \le n} \{u_{1i}^l + w_{ij}\}.$$

若 $u_{1j}^l > u_{1k}^l + w_{kj}$, 置 $u_{1j}^{l+1} \leftarrow u_{1k}^l + w_{kj}, J_j \leftarrow k$; 否则 $u_{1j}^{l+1} \leftarrow u_{1j}^l$, 且 J_j 不变.

步 2 若 $l + 1 = n - 1$, 进行第 3 步, 否则置 $l \leftarrow l + 1$, 返回第 1 步.

步 3 根据记录 $\boldsymbol{R}$ 找出自顶点 1 到其余各点的最短路.

例 3 求图1-3中自顶点 1 到各顶点的最短路.

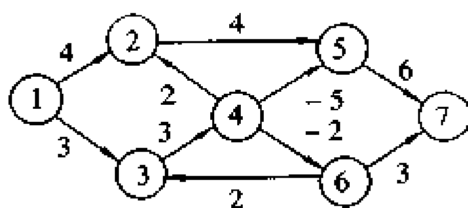


图 1-3

步 0 $u_{11}^1 = 0, u_{12}^1 = 4, u_{13}^1 = 3, u_{1j}^1 = +\infty, j = 4,5,6,7, \boldsymbol{R} = (1,1,1,1,1,1,1)$.

步 1 $u_{14}^1 > u_{13}^1 + w_{34} = 6, u_{15}^1 > u_{12}^1 + w_{25} = 8$; 故置 $u_{14}^2 \leftarrow 6, u_{15}^2 \leftarrow 8$; 其余 $u_{11}^2 = 0, u_{12}^2 = 4, u_{13}^2 = 3, u_{1j}^2 = +\infty, j = 6,7; J_4 \leftarrow 3, J_5 \leftarrow 2$, 其余不变.

由于 $l = 2 < n - 1 = 6$, 返回步 1.

重复上述过程, 6 次迭代后得 $u_{12}^6 = 4, u_{13}^6 = 3, u_{14}^6 = 3, u_{15}^6 = 8, u_{16}^6 = 1, u_{17}^6 = 4; \boldsymbol{R} = (1,1,1,5,2,4,6)$.

步 3 找出自顶点 1 到各顶点的最短路. 如 1 到 7 的最短路由 $\boldsymbol{R}$ 得

$$① \longrightarrow ② \longrightarrow ⑤ \longrightarrow ④ \longrightarrow ⑥ \longrightarrow ⑦$$

其长度 $u_{17}^6 = 4$.

这个算法的主要计算量是在第 1 步. 对每一个 j 要做 $n - 1$ 次加法和 $n - 1$ 次比较, 而 j 有 $n - 1$ 个. 故第 1 步计算量为 $O(n^2)$. 由于第 1 步最多重复 $n - 1$ 次, 所以该算法的计算量为 $O(n^3)$.

1.5 所有点对间的最短路算法

前面所介绍的算法都是求网络中自一个顶点到其余各顶点的最短路. 不难看出 1.4 节中的方法重复 n 次, 就可求出所有点对间的最短路. 这样做的计算量为 $O(n^4)$. 然而有更好的算法.

设 $\boldsymbol{C} = [c_{ij}], \boldsymbol{F} = [f_{ij}]$ 分别为 $p \times q$ 和 $q \times r$ 的实矩阵, 定义矩阵乘法如下:

$$\boldsymbol{P} = [p_{ij}] = \boldsymbol{C} \otimes \boldsymbol{F}.$$

$\boldsymbol{P}$ 为 $p \times r$ 阶矩阵, $p_{ij} = \min\limits_{1 \le k \le q} \{c_{ik} + f_{kj}\}$, 即用取最小代替通常的加法, 而用加法代替通常的乘法. 例如

$$C=\begin{bmatrix}2&3\\1&5\end{bmatrix},\quad F=\begin{bmatrix}3&2\\4&1\end{bmatrix},$$

$$P=C\otimes F=\begin{bmatrix}\min\{2+3,3+4\} & \min\{2+2,3+1\}\\ \min\{1+3,5+4\} & \min\{1+2,5+1\}\end{bmatrix}=\begin{bmatrix}5&4\\4&3\end{bmatrix}.$$

如果用新定义的矩阵乘法，那么上节中的算法可重新表示：

置 $l=1,(u_{11}^{l},u_{12}^{l},\cdots,u_{1n}^{l})=(0,w_{12},\cdots,w_{1n})$，

$$u_{1j}^{l+1}=(u_{11}^{l},u_{12}^{l},\cdots,u_{1n}^{l})\otimes\begin{bmatrix}w_{1j}\\w_{2j}\\\vdots\\w_{nj}\end{bmatrix}=\min_{1\leqslant k\leqslant n}\{u_{1k}^{l}+w_{kj}\},$$

因此，

$$(u_{11}{}^{l+1},u_{12}{}^{l+1},\cdots,u_{1n}{}^{l+1})=(u_{11}^{l},u_{12}^{l},\cdots,u_{1n}^{l})\otimes\begin{bmatrix}w_{11}&w_{12}&\cdots&w_{1n}\\w_{21}&w_{22}&\cdots&w_{2n}\\\vdots&\vdots&&\vdots\\w_{n1}&w_{n2}&\cdots&w_{nn}\end{bmatrix},$$

其中 $w_{ii}=0$；若 $(i,j)\in A$，则 $w_{ij}=+\infty$。

不难看出，这种表示法可用于求所有点对之间的最短路.

1.5.1　求所有点对之间的最短路的矩阵算法

步 0　置 $m\leftarrow 1$，

$$U^{m}=\begin{bmatrix}u_{11}^{m}&u_{12}^{m}&\cdots&u_{1n}^{m}\\u_{21}^{m}&u_{22}^{m}&\cdots&u_{2n}^{m}\\\vdots&\vdots&&\vdots\\u_{n1}^{m}&u_{n2}^{m}&\cdots&u_{nn}^{m}\end{bmatrix}=\begin{bmatrix}w_{11}&w_{12}&\cdots&w_{1n}\\w_{21}&w_{22}&\cdots&w_{2n}\\\vdots&\vdots&&\vdots\\w_{n1}&w_{n2}&\cdots&w_{nn}\end{bmatrix}=W$$

$$R=\begin{bmatrix}r_{11}&r_{12}&\cdots&r_{1n}\\r_{21}&r_{22}&\cdots&r_{2n}\\\vdots&\vdots&&\vdots\\r_{n1}&r_{n2}&\cdots&r_{nn}\end{bmatrix},\quad r_{ij}=j,\quad i,j=1,2,\cdots,n.$$

步 1　计算

$$U^{2m}=U^{m}\otimes U^{m}.$$

计算过程中若出现

$$u_{ij}^{m}>\min_{1\leqslant l\leqslant n}\{u_{il}^{m}+u_{lj}^{m}\}=u_{ik}^{m}+u_{kj}^{m},$$

则置 $r_{ij}\leftarrow k$.

步 2　若 $2m<n-1$，置 $m\leftarrow 2m$，返回步 1；否则进行步 3.

步 3　利用 $\boldsymbol{R}$ 找出所有点对间的最短路. 此时 $\boldsymbol{R}$ 中元素 r_{ij} 表示 i 到 j 的最短路必通过顶点 r_{ij}. 利用这一信息可找出 i 到 j 的最短路.

注意，此算法需迭代 $K=\{\log_2(n-1)\}$ 次，即要作 K 次矩阵乘法. 每次矩阵乘

法的计算量为 $O(n^3)$. 因此该算法的计算复杂性为 $O(n^3\log_2 n)$.

例 4 求图1-4中点对间的最短路.

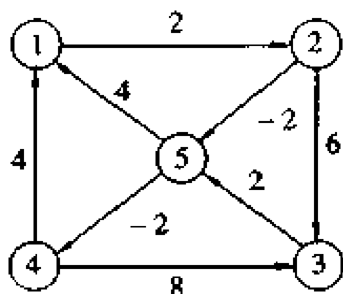


图 1-4

步 0

$$U^1 = W = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty \\ \infty & 0 & 6 & \infty & -2 \\ \infty & \infty & 0 & \infty & 2 \\ 4 & \infty & 8 & 0 & \infty \\ 4 & \infty & \infty & -2 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}.$$

步 1

$$U^2 = \begin{bmatrix} 0 & 2 & 8 & \infty & 0 \\ 2 & 0 & 6 & -4 & -2 \\ 6 & \infty & 0 & 0 & 2 \\ 4 & 6 & 8 & 0 & 10 \\ 2 & 6 & 6 & -2 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 2 & 2 & 4 & 2 \\ 5 & 2 & 3 & 5 & 5 \\ 5 & 2 & 3 & 5 & 5 \\ 1 & 1 & 3 & 4 & 3 \\ 4 & 1 & 4 & 4 & 5 \end{bmatrix}.$$

$$U^4 = \begin{bmatrix} 0 & 2 & 6 & -2 & 0 \\ 0 & 0 & 4 & -4 & -2 \\ 4 & 6 & 0 & 0 & 2 \\ 4 & 6 & 8 & 0 & 4 \\ 2 & 4 & 6 & -2 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 2 & 5 & 2 & 2 \\ 4 & 2 & 4 & 5 & 5 \\ 4 & 4 & 3 & 5 & 5 \\ 1 & 1 & 3 & 4 & 1 \\ 4 & 1 & 4 & 4 & 5 \end{bmatrix}.$$

步 2 找出所有点对间的最短路. 例如顶点 3 到顶点 2 的最短路. 由于 $r_{32} = 4$, $r_{34} = 5$, $r_{35} = 5$, $r_{42} = 1$, $r_{41} = 1$, $r_{12} = 2$, 故 3 到 2 的最短路为 ③ → ⑤ → ④ → ① → ②, 长度为 $u_{32}^4 = 6$.

矩阵乘法在计算程序上是简单的, 但运算量还比较大. 下面介绍的福劳得(K.W.Floyd)算法, 其计算复杂性为 $O(n^3)$.

1.5.2 求所有点对之间最短路的福劳得算法

设 u_{ij}^m 表示自顶点 i 到 j 且不经过顶点 $m, m+1, \cdots, n$ 的所有路中最短一条路的长度. 因此自顶点 i 到顶点 j 不经过顶点 $m+1, m+2, \cdots, n$ 的最短路有两种可能的情形: 或者它不经过 m 点, 此时 $u_{ij}^{m+1} = u_{ij}^m$; 或者经过 m 点, 此时 $u_{ij}^{m+1} = u_{im}^m + u_{mj}^m$. 因此, 总有

$$u_{ij}^{m+1} = \min\{u_{ij}^m, u_{im}^m + u_{mj}^m\}.$$

由 u_{ij}^{m} 的定义易知，u_{ij}^{n+1} 即为网络中自 i 到 j 的最短路长度.

步0 置 $m \leftarrow 1, u_{ij}^{m} = w_{ij} \quad 1 \leqslant i,j \leqslant n$，其中 $w_{ii} = 0, i = 1,2,\cdots,n$；当 $(i,j) \notin A$ 且 $i \neq j$ 时，$w_{ij} = +\infty$. $R = [r_{ij}], r_{ij} = j, 1 \leqslant i,j \leqslant n$.

步1 对一切 $i、j, 1 \leqslant i \neq j \leqslant n$，计算

$$u_{ij}^{m+1} = \min\{u_{ij}^{m}, u_{im}^{m} + u_{mj}^{m}\}.$$

若 $u_{ij}^{m} > u_{im}^{m} + u_{mj}^{m}$，则置 $r_{ij} \leftarrow m$.

步2 若 $m = n$，进行第3步；否则，置 $m \leftarrow m + 1$ 返回步1.

步3 由 R 找出所有点对间的最短路，其方法与矩阵算法中步3相同.

值得注意的是，当网络中有负回路时，计算过程中必出现某些 $u_{ii}^{m} < 0$，因此所有点对间的最短路算法，可用于探测网络中是否有负回路.

1.6 最短路的变种

1.6.1 最大可靠路

设 $N(V,A;Q)$ 为通信网络，$(i,j) \in A, 0 < q_{ij} \leqslant 1$ 表示弧 (i,j) 的可靠度. 一条路的可靠度定义为该路上各条弧的可靠度的乘积. 自一顶点 i 到另一顶点 j 的所有路中，可靠度最大的路称为 i 到 j 的最可靠路.

若令 $w_{ij} = -\mathrm{lb}\, q_{ij}, (i,j) \in A$，则 $N(V,A;Q)$ 中最可靠路等价于网络 $N(V,A;w)$ 中的最短路. 因此求最可靠路可化为求最短路.

1.6.2 最大容量路

设 $N(V,A;c)$ 是一有向网络，每一弧 $(i,j) \in A$ 有一个容量 $c_{ij} > 0$. 一条路 P 的容量定义为该路上弧的容量之最小者，即 P 的容量为

$$c(P) = \min_{(i,j)\in P}\{c_{ij}\}.$$

求 $N(V,A;c)$ 中自顶点1到顶点 n 的最大容量路与求最短路类似，只要把迪克斯特拉算法中的 min 用 max 代替，而加法用 min 代替即可. 具体叙述如下：

步0 置 $S = \{1\}, T = \{2,3,\cdots,n\}, R = (J_1, J_2, \cdots, J_n)$，其中 $J_i = 1, i = 1, 2, \cdots, n$；$u_1 = 0, u_j = c_{1j}, j \in T$. 注意，若 $(i,j) \notin A$ 有 $c_{ij} = 0$.

步1 在 T 中找一点 k，使得

$$u_k = \max_{j \in T}\{u_j\}.$$

置 $S \leftarrow S \cup \{k\}, T \leftarrow T - \{k\}$. 若 $T = \varnothing$，进行步3.

步2 对每一顶点 $j \in T$，修正 u_j 如下：

若 $u_j < \min\{u_k, c_{kj}\}$，置 $u_j \leftarrow \min\{u_k, c_{kj}\}, J_j \leftarrow k$. 然后返回步1.

步3 由 R 找出自顶点1到其余各顶点的最大容量路.

1.6.3 最大期望容量路

设 $N(V,A;Q,c)$ 是一个有向网络，又每一条弧 $(i,j) \in A$ 有一个可靠度 q_{ij} 和

一个容量 c_{ij},其中 $0 < q_{ij} \leqslant 1, c_{ij} > 0$. N 中一条路 P 的期望容量 $f(P)$ 定义为

$$f(P) = (\prod_{(i,j)\in P} q_{ij}) \min_{(i,j)\in P} \{c_{ij}\}.$$

自顶点 i 到顶点 j 的所有路中,期望容量最大的路称为 i 到 j 的**最大期望容量路**. 下面叙述求 N 中自点 1 到点 n 的最大期望容量路的算法.

步 0 置 $L \leftarrow \varnothing, f(L) = 0$.

步 1 求出 N 中自点 1 到点 n 的最大可靠路 P.

若 N 中不存在自 1 到 n 的路,终止,L 为所求;否则计算 $f(P)$.

若 $f(P) > f(L)$,置 $L \leftarrow P, f(L) \leftarrow f(P)$.

步 2 (修改网络 N) 计算 P 的容量 $c(P)$.

$$c(P) = \min_{(ij)\in P} \{c_{ij}\}.$$

置 $A \leftarrow A - \{(i,j) \in A \mid c_{ij} \leqslant c(P)\}$,用新的网络代替原网络返回步 1.

这个算法最多迭代 $m = |A|$ 次,每一次要求一条最可靠路,其计算量为 $O(n^2)$. 因此该算法总的计算量为 $O(mn^2)$.

2 最大流问题

2.1 最大流的基本概念

设 $N(V,A;c)$ 为一有向网络,对每一条弧 $(i,j) \in A$,有一个非负实数 c_{ij} 称为弧 (i,j) 的容量或通过能力. 设 s 和 t 是 V 中两个指定的顶点,分别称为发点和收点,称这样的网络为运输网络,记为 $N = (s,t,V,A;c)$. 假设对任意 i、$j \in V$, i 到 j 最多有一条弧,则称网络 N 是简单的.

运输网络 N 上的一个流定义为 A 上的一个函数 f,使得

1° 对任意 $(i,j) \in A$, $0 \leqslant f(i,j) \leqslant c_{ij}$.

2° 对任意 $i \in V \setminus \{s,t\}$,

$$\sum_{k\in\alpha(i)} f(i,k) = \sum_{k\in\beta(i)} f(k,i).$$

称 $f(i,j)$ 为弧 (i,j) 上的流量. 由 s 净流出的数量称为流 f 的值,记为 $v(f)$,即

$$v(f) = \sum_{k\in\alpha(s)} f(s,k) - \sum_{k\in\beta(s)} f(k,s).$$

使流值达到最大的流,称为 N 的**最大流**. 显然,由流的定义可推出,N 上任何一个流 f 必有

$$\sum_{k\in\alpha(s)} f(s,k) - \sum_{k\in\beta(s)} f(k,s) = \sum_{k\in\beta(t)} f(k,t) - \sum_{k\in\alpha(t)} f(t,k).$$

因此,N 上 s 到 t 的最大流问题,可描述为下述线性规划:

$$\begin{cases} \max v, \\ \sum\limits_{k\in\alpha(i)} f(i,k) - \sum\limits_{k\in\beta(i)} f(k,i) = \begin{cases} v, & i = s, \\ 0, & i \in V - \{s,t\}, \\ -v, & i = t, \end{cases} \\ 0 \leqslant f(i,j) \leqslant c_{ij}, (i,j) \in A. \end{cases} \tag{2-1}$$

设 $X \subset V, \bar{X} = V - X$,并且 $s \in X, t \in \bar{X}$,则称

$$(X,\bar{X}) = \{(i,j) \in A \mid i \in X, j \in \bar{X}\}$$

为分离 s 和 t 的割集,或简称为 s-t 割. s-t 割$(X,\bar{X})$的割量定义为

$$c(X,\bar{X}) = \sum_{(i,j)\in(X,\bar{X})} c_{ij}.$$

容易证明,N 上的任何一个流 f 及任何一个 s-t 割$(X,\bar{X})$,均有

$$v(f) = \sum_{(i,j)\in(X,\bar{X})} f(i,j) - \sum_{(j,i)\in(\bar{X},X)} f(j,i) \leqslant \sum_{(i,j)\in(X,\bar{X})} c_{ij} = c(X,\bar{X}). \tag{2-2}$$

设 $\mathscr{F}$ 为(2-1)式的所有可行解集合,$\mathscr{D}$ 为所有 s-t 割集合,则有以下著名的福特-冯克孙(Ford-Fulkerson)最大流最小割定理:

定理 1　$\max\limits_{f\in\mathscr{F}} v(f) = \min\limits_{(X,\bar{X})\in\mathscr{D}} c(X,\bar{X})$.

最大流问题(2-1)是一个线性规划问题,因此可以用单纯形算法或内点法求最大流.然而由于问题(2-1)的特殊性,也可以利用网络方法求解.

2.2　最大流算法

求解最大流问题有多个算法,下面叙述有影响的4种算法:

2.2.1　福特-冯克孙算法

步0　$N = (s,t,V,A;c)$ 是一运输网络,f 是 N 上的一个流(开始可取 $f \equiv 0$).

步1　构造关于流 f 的剩余网络 $N(f) = (s,t,V,\tilde{c};\tilde{c})$,其中 $\tilde{c} = A_1 \cup A_2$,

$$A_1 = \{(i,j) \mid (i,j) \in A \text{ 且 } f(i,j) < c_{ij}\},$$

$$A_2 = \{(i,j) \mid (j,i) \in A \text{ 且 } f(j,i) > 0\},$$

$$\tilde{c}_{ij} = \begin{cases} c_{ij} - f(i,j), & \text{若}(i,j) \in A_1, \\ f(j,i), & \text{若}(i,j) \in A_2. \end{cases}$$

步2　在 $N(f)$ 中求 s 到 t 的路.

(2-1)　若 $N(f)$ 中不存在 s 到 t 的路,算法终止.f 为 N 中最大流.

(2-2)　设 P 是 $N(f)$ 中自 s 到 t 的路,计算 P 的容量 θ:

$$\theta = \min\{\tilde{c}_{ij} \mid (i,j) \in P\}.$$

步 3 改进流 f,

$$\tilde{c}(i,j)=\begin{cases}f(i,j)+\theta, & 若(i,j)\in P\cap A_1,\\ f(i,j)-\theta, & 若(j,i)\in P\cap A_2,\\ f(i,j), & 其余.\end{cases}$$

置 $f\leftarrow\tilde{f}$,返回步 1.

福特 - 冯克孙算法不仅不是多项式算法,而且有例子说明,当容量为无理数时可能不收敛于最大流.但每次迭代最大流的值是严格增加的,因此当容量 c 为整函数时,该算法在有限步内得到最大流.

针对福特 - 冯克孙算法的弱点,艾德门兹(Edmonds)和卡普(Karp)在 1972 年对上述算法进行了简单修正,即只需把上述算法中的第 2 步求 $N(f)$ 中自 s 到 t 的路,改为求 $N(f)$ 中自 s 到 t 的最短(弧数最小的)路即可.修正后的算法不仅对无理数容量适用,而且其计算复杂性为 $O(nm^2)$,其中 $m=|A|$.

2.2.2 狄尼茨算法

早在 1970 年,狄尼茨(Dinits)提出了与艾德门兹和卡普的算法思想相类似的方法,不同之处在于狄尼茨算法是在 $N(f)$ 中找出 s 到 t 的所有最短(弧数最少)路(而不是只找一条),然后在这些最短路组成的网络($N(f)$ 的子网络)中,用深探法求出这个子网络的极大流(不必是最大流).用这个极大流修改当前流 f,从而得到 N 上的一个新流 $\tilde{f}$.容易证明 $N(\tilde{f})$ 中 s 到 t 的最短路长度严格大于 $N(f)$ 中 s 到 t 的最短路长度.由此可以估计出狄尼茨算法的迭代步数不会超过 $n-1$ 次.

狄尼茨最大流算法:

步 0 设 $N=(s,t,V,A;c)$ 为运输网络,f 为 N 上的一个流(开始可取 $f\equiv 0$).

步 1 构造关于 f 的剩余网络 $N(f)$(同福特 - 冯克孙算法第 1 步).

步 2 由 $N(f)$ 构造分层网络 $\mathrm{AN}(f)$.由迪克斯特拉算法(见 1.2 节)求出 $N(f)$ 中自 s 到每一点 j 的最短路长度 u_j(这里路的长度指路上弧的个数),$N(f)=(s,t,V,\tilde{A};\tilde{c})$.

(2-1) 若 $u_t=\infty$,即 $N(f)$ 不存在 s 到 t 的路,算法终止.f 是 N 中最大流.

(2-2) 设 $u_t=l<\infty$,定义

$$V_0=\{s\},\quad V_i=\{j\mid u_j=i\}(i=1,\cdots,l-1),\quad V_l=\{t\},$$

$$A^*=\bigcup_{k=0}^{l-1}\{(i,j)\mid i\in V_k,j\in V_{k+1},(i,j)\in\tilde{A}\}$$
$$=A^*\cap A_1\cup A^*\cap A_2,$$

$$c_{ij}^*=\tilde{c}_{ij},\quad (i,j)\in A^*,\quad V^*=\bigcup_{l=0}^{l}V_k.$$

$$\mathrm{AN}(f)=(s,t,V^*,A^*;c^*),$$

称 $\mathrm{AN}(f)$ 为 $N(f)$ 的分层网络.

步 3 求 $\mathrm{AN}(f)$ 中自 s 到 t 的极大流.

(3-0) 设 g 为 AN(f) 的初始流. 这儿取 $g \equiv 0$.

(3-1) 求 AN(f) 中的路.

(3-1-0) 给 s 标号(ϕ),称其为当前标号.

(3-1-1) 设顶点 i 的当前标号为 $l(i)$,且 $i \in V_k, 0 \leqslant k \leqslant l-1$,在 V_{k+1} 中找一点 j,使 $(i,j) \in A^*$.

若 V_{k+1} 中不存在这样的 j,则当 $i = s$ 时转(3-4);当 $i \neq s$ 时转(3-3). 否则,设 $j \in V_{k+1}$,当 $j = t$ 时转(3-2),否则给 j 标号 $l(j) = i$. 然后置 $i \leftarrow j$,返回(3-1-1).

(3-2) 利用标号返回追踪,找出 s 到 t 的路 P. 计算

$$\theta = \min\{c_{ij}^* \mid (i,j) \in P\}.$$

置

$$g_{ij} \leftarrow \begin{cases} g_{ij} + \theta, & 若(i,j) \in P, \\ g_{ij}, & 其余, \end{cases}$$

$$c_{ij}^* \leftarrow \begin{cases} c_{ij}^* - \theta, & 若(i,j) \in P, \\ c_{ij}^*, & 其余, \end{cases}$$

$$A^* \leftarrow A^* - \{(i,j) \mid (i,j) \in P 且 c_{ij}^* = 0\},$$

$$\mathrm{AN}(f) \leftarrow (s, t, V^*, A^*; c^*).$$

抹去 AN(f) 中所有顶点标号,返回(3-1).

(3-3) 由于在 AN(f) 中,自顶点 i 无发出去的弧,所以 AN(f) 中自 s 到 t 的一切路均不过顶点 i. 此时可从 AN(f) 中去掉顶点 i,或者置

$$A^* \leftarrow A^* - \{(r,i) \mid r \in V_{k-1} 且 (r,i) \in A^*\}.$$

在新的网络中,用顶点 $l(i)$ 替代 i,即置

$$i \leftarrow l(i).$$

返回(3-1-1).

(3-4) 由于 $i = s$ 且自 s 无发出弧,从而 AN(f) 中不存在自 s 到 t 的路.

步 4 改进 N 上的流 f 如下:

置

$$f_{ij} \leftarrow \begin{cases} f_{ij} + g_{ij}, & 若(i,j) \in A_1^*, \\ f_{ij} - g_{ji}, & 若(j,i) \in A_2^*, \\ f_{ij}, & 其余. \end{cases}$$

然后返回步 1.

狄尼茨算法的计算复杂性分析如下:构造分层网络其复杂性为 $O(n^2)$,求分层网络的极大流的复杂性为 $O(nm)$. 由于迭代次数最多为 $n-1$,所以狄尼茨算法的计算复杂性最多为 $O(n^2 m)$.

2.2.3 卡扎诺夫算法

卡扎诺夫(Kazanov)1974 年提出的最大流算法,仍是目前的最好算法. 其算法的基本思路与狄尼茨算法基本相同,只是在求分层网络 AN(f) 的极大流时采用不

同的方法.狄尼茨算法是用"深探法"求 AN(f) 中 s 到 t 的路,因此求一条路的复杂性为 $O(n)$.由于每找到一条路去掉至少一条弧,故求 AN(f) 的极大流的复杂性为 $O(nm)$.而卡扎诺夫算法在求分层网络 AN(f) 的极大流时采用"推拉"(push-pull)方法,其计算复杂性为 $O(n^2)$.这样一来,求 N 的最大流的计算复杂为 $O(n^3)$,它好于狄尼茨算法.

下面只叙述卡扎诺夫算法中如何求 AN(f) 上的极大流,即只需改动狄尼茨算法中的第3步.为此先引进一个定义.网络 AN(f) = $(s,t,V^*,A^*;c^*)$ 中任意一个顶点 $i\in V^*$ 的容量 $c(i)$ 定义为

$$c(i) = \min\left\{\sum_{j\in\alpha(i)} c_{ij}^*, \sum_{j\in\beta(i)} c_{ji}^*\right\}, i\neq s,t,$$

$$c(s) = \sum_{j\in\alpha(s)} c_{sj}^*,\quad c(t) = \sum_{j\in\beta(t)} c_{jt}^*.$$

步3　求 AN(f) 中 s 到 t 的极大流.

(3-0)　设 g 是 AN(f) = $(s,t,V^*,A^*;c^*)$ 的一个初始流(开始取 $g\equiv 0$). $Q_s\leftarrow\varnothing, Q_t\leftarrow\varnothing$ 队列记号,满足先进先出.

(3-1)　计算 AN(f) 中各顶点的通过能力 $c(i)$.设

$$c(k) = \min\{c(i) \mid i\in V^*\}.$$

若 $c(k)=0$,进行(3-1-1);否则进行(3-1-2).

(3-1-1)　若 $k=s$ 或 t,转第4步;否则置

$$V^*\leftarrow V^* - \{k\};$$

$$A^*\leftarrow A^* - \bigcup_{j\in\alpha(k)}\{(k,j)\} - \bigcup_{j\in\beta(k)}\{(j,k)\},$$

$$\text{AN}(f)\leftarrow(s,t,V^*,A^*;c^*).\text{返回(3-1)}.$$

(3-1-2)　置 $Q_t\leftarrow k, Q_s\leftarrow k, r(k)\leftarrow c(k)$;对一切 $u\neq k, r(u)\leftarrow 0$.

(3-2)　前推(push)

按先进先出原则,取 $u\in Q_t$,若 $u=t$,转(3-3);置 $Q_t\leftarrow Q_t-u$.

(3-2-1) 在 AN(f) 中任取一弧 (u,i),置

$$\delta=\min\{c_{ui}^*, r(u)\}, c_{ui}^*\leftarrow c_{ui}^*-\delta, r(u)\leftarrow r(u)-\delta;$$

$$g_{ui}\leftarrow g_{ui}+\delta, r(i)\leftarrow r(i)+\delta, Q_t\leftarrow Q_t+i.$$

(3-2-2)　若 $c_{ui}^*=0$,置 $A^*\leftarrow A^*-\{(u,i)\}$;

若 $r(u)>0$,返回(3-2-1);否则回到(3-2).

(3-3)　后拉(pull)

按先进先出原则,取 $u\in Q_s$,若 $u=s$,转第4步;否则置 $Q_s\leftarrow Q_s-u$.

(3-3-1)　在 AN(f) 中任取一弧 (i,u),置

$$\delta=\min\{c_{iu}^*, r(u)\}, c_{iu}^*\leftarrow c_{iu}^*-\delta, r(u)\leftarrow r(u)-\delta;$$

$$g_{i,u}\leftarrow g_{iu}+\delta, r(i)\leftarrow r(i)+\delta, Q_s\leftarrow Q_s+i.$$

(3-3-2)　若 $c_{iu}^*=0$,置 $A^*\leftarrow A^*-\{(i,u)\}$;

若 $r(u)>0$,返回(3-3-1);否则回到(3-3).

当运输网络 $N=(s,t,V,A;c)$ 满足对每一条弧 $(i,j)\in A$ 有 $c_{ij}=1$ 时,则称 N 为**单位容量网络**.单位容量网络上的最大流问题,在某些纯组合问题里很有用

途,如二部图的最大匹配等.卡扎诺夫算法用于求单位容量网络上的最大流,其计算复杂性为 $O(mn^{2/3})$;当单位容量网络是二部图时,其计算复杂性为 $O(mn^{1/2})$.

2.3　具有上下界容量网络的最大流

如果运输网络中每一条弧 (i,j) 不仅有一个容量的上界 c_{ij},而且还有一个下界 l_{ij}.设这个网络为 $N=(s,t,V,A;l,c)$,若映射 $f:A\to\mathbf{R}$ 满足:

1° 对一切 $(i,j)\in A$, $l_{ij}\leqslant f(i,j)\leqslant c_{ij}$;

2° 对每一个 $i\in V-\{s,t\}$ 有

$$\sum_{j\in\alpha(i)}f(i,j)=\sum_{j\in\beta(i)}f(j,i);$$

则称 f 为 N 上的流.使 $\sum_{j\in\alpha(s)}f(s,j)-\sum_{j\in\beta(s)}f(j,s)$ 达到最大的流,称为 N 上的最大流.

求有上下界容量网络的最大流,其算法与求一般运输网络的最大流算法基本相同,只需做两点修正:

(1) 需要给出 N 上的第一个初始流 f,而在一般运输网络中 $f=0$ 即可.

(2) 当有了流 f 后,在构造 $N(f)$ 时,A_2 的定义需修改为

$$A_2=\{(i,j)\mid(j,i)\in A\text{ 且 }f(j,i)>l_{ji}\};$$

$$c_{ij}^*=f(j,i)-l_{ji},\quad(i,j)\in A_2.$$

因此,关键在于找出 N 上的一个初始流.现叙述一个方法如下:

设 $N=(s,t,V,A;l,c)$ 为有上下界容量的运输网络.构造一个辅助网络 $\widetilde{N}=(s^*,t^*,\widetilde{V},\widetilde{A};c)$,其中

$$\widetilde{V}=V\cup\{s^*,t^*\},$$

$$\widetilde{A}=A\cup\{(s^*,i)\mid i\in V\}\cup\{(i,t^*)\mid i\in V\}\cup\{(s,t),(t,s)\},$$

$$\tilde{c}_{ij}=\begin{cases}c_{ij}-l_{ij}, & \text{若}(i,j)\in A,\\ \sum\limits_{k\in\beta(j)}l_{kj}, & \text{若 } i=s^*,j\in V,\\ \sum\limits_{k\in\alpha(i)}l_{ik}, & \text{若 } j=t^*,i\in V,\\ \infty, & \text{若 } i,j\in\{s,t\}\text{ 且 } i\neq j.\end{cases}$$

容易证明,当且仅当 $\widetilde{N}$ 中自 s^* 到 t^* 的最大流的值为 $\sum_{i\in V}\sum_{j\in\beta(i)}l_{ji}$ 时,N 中存在满足容量上下界的流.由于 $\widetilde{N}$ 是一般运输网络,故可以用上一节的方法求解.

设 $\tilde{f}$ 是 $\widetilde{N}$ 上的最大流,其值 $v(\tilde{f})=\sum_{i\in V}\sum_{j\in\beta(i)}l_{ji}$,那么对一切 $(i,j)\in A$,定义 $f(i,j)=\tilde{f}(i,j)+l_{ij}$,则 f 是 N 上的流且满足 $l_{ij}\leqslant f(i,j)\leqslant c_{ij}$,$(i,j)\in A$.有了 N 上的流 f,就可以求 N 上的最大流了.

最大流问题均可以化为循环流问题.所谓循环流乃指如下模型:设 $N(V,A;l,c)$ 为一有向网络,$f:A\to\mathbf{R}$ 的映射,并且满足:

1° 对任意$(i,j)\in A$,有 $l_{ij}\leqslant f(i,j)\leqslant c_{ij}$;

2° 平衡条件:

$$\sum_{j\in\alpha(i)}f(i,j)-\sum_{j\in\beta(i)}f(j,i)=0,\quad i\in V;$$

则称 f 是 N 上的**一个可行循环流**.

最大循环流是指求一个循环流 f,在 f 满足 1° 和 2° 下,使 $\sum_{(i,j)\in A}f(i,j)$ 最大.

前述的 s 到 t 的最大流问题,也可以化为循环流问题.这只要在 N 中增加一条弧(t,s)(这里假设原网络中不存在(t,s) 弧),取 $c_{ts}=\infty$ 而 $l_{ts}=-\infty$,并且目标函数取为 $\max f(t,s)$ 即可.

循环流问题自身也有很多应用,这里只举关于三角债问题的一个例子(该问题最早由山东曲阜师大马克杰等提出的,由李国君化为最大流).

设 $N(V,A;c)$ 是一个有向网络,V 中的顶点 i 代表第 i 个企业,弧$(i,j)\in A$ 表示第 i 个企业拖欠第 j 个企业债务,c_{ij} 表示第 i 个企业拖欠第 j 个企业的债务量.解决企业间的三角债问题的一种简便的方法是相互抵消,以减少总债务量.例如甲欠乙 3 万元,乙欠丙 2 万元、丙欠甲 5 万元,那么他们相互抵消后,总债务量由 10 万元减少为 4 万元,即甲欠乙 1 万元,丙欠甲 3 万元.

一般地,设 Q 为 N 中的有向圈,记

$$q=\min_{(i,j)\in Q}\{c_{ij}\},$$

则在 Q 上的抵消量为 $|Q|q$,其中 $|Q|$ 为有向圈(亦称回路)Q 中弧的个数.问题是如何进行抵消,使抵消总量最大,或者说剩余的债务量最小.这个问题正是最大循环流问题:

$$\begin{cases}\max\sum_{(i,j)\in A}f(i,j),\\ \sum_{j\in\alpha(i)}f(i,j)-\sum_{j\in\beta(i)}f(j,i)=0,\quad i\in V,\\ 0\leqslant f(i,j)\leqslant c_{ij}.\end{cases}\tag{2-3}$$

下一章将详细讨论它的解法.

2.4　可行性定理及其应用

设 $N(V,A;l,c)$ 是一个具有上下界容量的有向网络,那么 N 中是否存在循环流 f?即下述问题是否有解:

$$\begin{cases}\sum_{j\in\alpha(i)}f(i,j)-\sum_{j\in\beta(i)}f(j,i)=0,\quad i\in V,\\ l_{ij}\leqslant f(i,j)\leqslant c_{ij},\qquad (i,j)\in A.\end{cases}\tag{2-4}$$

可行性定理断言:问题(2-4) 有解,当且仅当对任意的 $X\subseteq V$,有 $c(X,\bar{X})\geqslant l(\bar{X},$

$X)$，其中 $\bar{X} = V - X$，

$$c(X,\bar{X}) = \sum_{(i,j)\in(X,\bar{X})} c_{ij},\quad l(\bar{X},X) = \sum_{(i,j)\in(\bar{X},X)} l_{ij}.$$

(1) 作为问题(2-4)的一种特殊而有用的情形，设 $N(V,A;c)$ 为有向网络，顶点集 V 划分为三部分：S、R、T，分别称它们为发点、中间点和收点集合．对每一 $i \in S$，有一个发量 a_i；每一点 $i \in T$ 有一个收量 b_i．问题是下述(2-5)式有解的条件是什么？

$$\begin{cases} \sum_{j\in\alpha(i)} f(i,j) - \sum_{j\in\beta(i)} f(j,i) \leqslant a_i, & i \in S, \\ \sum_{j\in\alpha(i)} f(i,j) - \sum_{j\in\beta(i)} f(j,i) = 0, & i \in R, \\ \sum_{j\in\alpha(i)} f(i,j) - \sum_{j\in\beta(i)} f(j,i) \geqslant b_i, & i \in T, \\ 0 \leqslant f(i,j) \leqslant c_{ij}, & (i,j) \in A. \end{cases} \tag{2-5}$$

其中 a_i、$b_j \geqslant 0, i \in S, j \in T$.

由可行性定理不难证明，(2-5)式有解，当且仅当对任意 $X \subseteq V$，有

$$c(X,\bar{X}) \geqslant b(T \cap \bar{X}) - a(S \cap \bar{X}), \tag{2-6}$$

其中 $b(T \cap \bar{X}) = \sum_{i\in T\cap\bar{X}} b_i$, $a(S \cap \bar{X}) = \sum_{i\in S\cap\bar{X}} a_i$.

由此结果可以推出著名的海尔(P. Hall)定理.

(2) 设 $T_1, T_2, \cdots, T_m$ 是有限集 $S = \{1,2,\cdots,n\}$ 的一个子集族，则该子集族有不同代表系，当且仅当对任意子集 $X \subseteq \{T_1, T_2, \cdots, T_m\}$，有

$$\left|\bigcup_{T_i\in X} T_i\right| \geqslant |X|. \tag{2-7}$$

(3) 设 $D = (V,A)$ 为一个有向图，对每一点 $v \in V$，用 $d_D^+(v)$ 和 $d_D^-(v)$ 分别表示 v 的出次和入次．对每一个顶点 $v \in V$，给定4个非负整数 $a(v)$, $a'(v)$, $b(v)$ 和 $b'(v)$ 使得 $a(v) \leqslant a'(v)$, $b(v) \leqslant b'(v)$，则 D 有一个子图 $H = (V, A_h)$，使得对每一点 $v \in V$ 有

$$a(v) \leqslant d_H^+(v) \leqslant a'(v),$$
$$b(v) \leqslant d_H^-(v) \leqslant b'(v),$$

当且仅当对每一个 $X \subseteq V$ 有

$$\begin{cases} a(X) \leqslant \sum_{y\in\alpha(X)} \min\{b'(y), |(X,y)|\}, \\ b(X) \leqslant \sum_{y\in\beta(X)} \min\{a'(y), |(y,X)|\}, \end{cases} \tag{2-8}$$

其中 $\alpha(X) = \bigcup_{i\in X}\alpha(i)$, $\beta(X) = \bigcup_{i\in X}\beta(i)$, $|(Y,Z)|$ 表示起点在 Y 里终点在 Z 里的弧之数目.

(4) 设 $a_i(i=1,2,\cdots,m)$ 和 $b_j(j=1,2,\cdots,n)$ 是非负整数，其中 $b_1 \geqslant b_2 \geqslant \cdots \geqslant b_n$，则存在一个 $m \times n$ 的 0-1 矩阵 $A=[a_{ij}]$ 满足对一切 $j=1,2,\cdots,n$，$i=1,2,\cdots,m$，

$$\sum_i a_{ij} \geqslant b_j, \quad \sum_j a_{ij} \leqslant a_i,$$

当且仅当对每一 $k=1,2,\cdots,n$，有

$$\sum_{j=1}^{k} b_j \leqslant \sum_{j=1}^{k} a_j^*, \tag{2-9}$$

其中

$$a_j^* = |\{i \mid a_i \geqslant j\}|.$$

3 最小费用流

最小费用流问题不仅在工农业及国防上有广泛的用途，它也是组合最优化的重要研究内容之一.

3.1 最小费用流的模型及解的性质

一般最小费用流问题的数学模型有下述几种形式：

设 $N=(V,A;b,l,c)$ 是有向网络，其中 V 是顶点集合，A 为弧集，b 为 A 上的费用函数，l 和 c 为 A 上的下界和上界容量函数. 求 N 上满足容量限制和供需要求的总费用最小的流.

设 $f(i,j)$ 为弧 $(i,j) \in A$ 上的流，最小费用流问题的第一种形式为，

$$\begin{cases} \min \sum_{(i,j)\in A} b_{ij} f(i,j), \\ -\sum_{j\in\alpha(i)} f(i,j) + \sum_{j\in\beta(i)} f(j,i) = a_i, \quad i \in V, \\ l_{ij} \leqslant f(i,j) \leqslant c_{ij}, \quad (i,j) \in A. \end{cases} \tag{3-1}$$

通常假定 b_{ij}、l_{ij}、c_{ij} 都为非负整数，而 a_i 为整数且 $\sum_{i\in V} a_i = 0$. 当 $a_i > 0$ 时，i 为收点，$a_i < 0$ 时 i 为发点，$a_i = 0$ 时 i 为中间点.

第二种形式为循环流问题，

$$(\mathrm{P}) \quad \begin{cases} \min \sum_{(i,j)\in A} b_{ij} f(i,j), \\ -\sum_{j\in\alpha(i)} f(i,j) + \sum_{j\in\beta(i)} f(j,i) = 0, \quad i \in V, \\ l_{ij} \leqslant f(i,j) \leqslant c_{ij}, \quad (i,j) \in A. \end{cases} \tag{3-2}$$

第三种形式为

$$\begin{cases}\min \sum_{(i,j)\in A} b_{ij} f(i,j), \\ -\sum_{j\in\alpha(i)} f(i,j) + \sum_{j\in\beta(i)} f(j,i) = \begin{cases} -v, & i = s, \\ 0, & i \in V - \{s,t\}, \\ v, & i = t, \end{cases} \\ l_{ij} \leqslant f(i,j) \leqslant c_{ij}, \qquad (i,j)\in A. \end{cases} \tag{3-3}$$

不难看出，(3-1) 式可化为(3-3) 式，(3-3) 式也可化为(3-2) 式. 无论哪种形式，它们都是线性规划问题. 根据线性规划的理论，最小费用循环流(P) 的对偶为

$$(\mathrm{D})\quad \begin{cases}\max \sum_{(i,j)\in A} (l_{ij}v_{ij} - c_{ij}u_{ij}), \\ \pi_j - \pi_i + v_{ij} - u_{ij} = b_{ij}, & (i,j)\in A, \\ v_{ij}, u_{ij} \geqslant 0, & (i,j)\in A. \end{cases}$$

(1) 由线性规划最优性互补条件得：$\{f(i,j)\}$ 是 (P) 的可行解，则$\{f(i,j)\}$ 是 (P) 的最优解，当且仅当存在顶点位势$\{\pi_i\}$，使得

$$\begin{cases} 若\ \pi_j - \pi_i < b_{ij}, & 则\ f(i,j) = l_{ij}, \\ 若\ \pi_j - \pi_i > b_{ij}, & 则\ f(i,j) = c_{ij}. \end{cases} \tag{3-4}$$

设$\{f(i,j)\}$ 是(P) 的一个可行解，那么由$\{f(i,j)\}$ 和网络 $N = (V,A;b,l,c)$ 可构造另一个网络 $N(f) = (V, \tilde{A}; \tilde{b}, \tilde{e}, \tilde{c})$，并称它为 N 关于$\{f(i,j)\}$ 的剩余网络. 其中 $\tilde{A} = A^+ \cup A^-$，而

$$\begin{cases} A^+ = \{(i,j)\in A \mid f(i,j) < c_{ij}\}, \\ A^- = \{(i,j) \mid (j,i)\in A\ 且\ f(j,i) > l_{ji}\}, \\ \tilde{b}_{ij} = \begin{cases} b_{ij}, & (i,j)\in A^+, \\ -b_{ji}, & (i,j)\in A^-, \end{cases} \\ \tilde{c}_{ij} = \begin{cases} c_{ij} - f(i,j), & (i,j)\in A^+, \\ f(j,i) - l_{ji}, & (i,j)\in A^-, \end{cases} \\ \tilde{l}_{ij} = 0 & (i,j)\in \tilde{A}. \end{cases} \tag{3-5}$$

(2) 设$\{f(i,j)\}$ 是(P) 的可行解，$N(f)$ 是 N 关于流$\{f(i,j)\}$ 的剩余网络，则$\{f(i,j)\}$ 是(P) 的最优解，当且仅当存在顶点位势$\{\pi_i\}$，使得对 $N(f)$ 中每一条弧(i,j) 有

$$\tilde{b}_{ij} - (\pi_j - \pi_i) \geqslant 0.$$

设 L 是$N(f)$ 中的一个有向圈，L 的权定义为L 中弧的费用之和，即

$$\tilde{b}(L) = \sum_{(i,j)\in L} \tilde{b}_{ij}.$$

若 $\tilde{b}(L) < 0$，则称 L 为负回路.

(3) (P) 的可行解$\{f(i,j)\}$ 是最优的，当且仅当 N 关于流$\{f(i,j)\}$ 的剩余网络 $N(f)$ 中不含负回路.

3.2 最小费用流的最小平均圈算法

这个算法主要应用 3.1 节中的(3).其思路是在有一个可行流 f 后,找出 $N(f)$ 中平均费用最小的回路.若它是负回路,则修正流 f,这样就得到一个更好的流;若它不是负回路,则由(3)知,它就是最优解.

3.2.1 最小平均圈的动态规划算法

对最小平均圈问题,1978 年卡普(Karp)提出了一个强多项式算法.

设 $N=(V,A;b)$ 是一个有向网络,L 是 N 中一个有向圈,L 的平均费用定义为 $b(L)/|L|=\frac{1}{|L|}\sum_{(i,j)\in L}b_{ij}$,其中 $|L|$ 表示 L 中弧的条数.最小平均圈就是在 N 的一切有向圈中,使 $b(L)/|L|$ 最小的圈.

有向网络中一条 k 迹是指由 k 条弧组成的路,但这 k 条弧不必不同.一条 k 迹的长度定义为 k 迹上 k 条弧的费用之和.自点 s 到点 j 的最短 k 迹,是指 s 到 j 的所有 k 迹中长度最短的那条 k 迹,并用 $d_s^{(k)}(j)$ 表示 s 到 j 的最短 k 迹的长度.

不失一般性,假设 N 是强连通(即任意两个顶点 u 和 v,存在 u 到 v 的有向路),s 是 N 中任一固定点,用 $d^{(k)}(j)$ 表示 s 到 j 的最短 k 迹的长度.最小平均费用圈的动态规划法:

步 0 置

$$d^{(0)}(j)=\begin{cases}0, & 若\ j=s,\\ \infty, & 若\ j\in V\ 且\ j\neq s.\end{cases}$$

步 1 对每一 $j\in V,k=1,2,\cdots,n$,计算

$$d^{(k)}(j)=\min\{d^{(k-1)}(i)+b_{ij}\mid i\in V,(i,j)\in A\}.$$

步 2 计算

$$\lambda^*=\min_{j\in V}\{\max_{0\leqslant k\leqslant n-1}\{(d^{(n)}(j)-d^{(k)}(j))/(n-k)\}\}. \tag{3-6}$$

设 $\lambda^*=(d^{(n)}(j_0)-d^{(r)}(j_0))/(n-r)$.

步 3 找出最小平均费用圈.

这个算法的有效性可分两种情况说明:

(1) $\lambda^*=0$.此时 N 中不含负回路,故可用 1.4 节中方法求出自 s 到各点的最短路.设 π_j 表示自点 s 到点 j 的最短路长度,则显然对每一点 $j\in V$ 有

$$\pi_j=\min_{0\leqslant k\leqslant n-1}\{d^{(k)}(j)\}\leqslant d^{(n)}(j),$$

从而

$$\max_{0\leqslant k\leqslant n-1}\{d^{(n)}(j)-d^{(k)}(j)\}\geqslant 0.$$

由第 2 步中 j_0 和 r 的选取及 $\lambda^*=0$,可知 $d^{(n)}(j_0)=d^{(r)}(j_0)=\pi_{j_0}$,由于 s 到 j_0 的 n 迹中必含回路,而 s 到 j_0 的最短路中不含回路,设 s 到 j_0 的 n 迹中有一条初等回路 L,则 $b(L)=0$.事实上,由于 N 中无负回路,故这条 n 迹中每一条回路的长度必等于 0,从而 L 也就是 N 中最小平均费用圈.

(2) $\lambda^* \neq 0$. 此时可化为情形(1). 将网络 N 中每一弧(i,j)的权 b_{ij}减去一个常数 θ,那么 N 中每一个有向圈 L 的平均费用减少 θ,而每一条 k 迹的费用为 $d_{(j)}^{(k)} - k\theta$,因此在新的权下,(3-6) 式为

$$\tilde{\lambda} = \lambda^* - \theta = \min_{j\in V}\max_{0\leqslant k\leqslant n-1}\left\{\frac{d_{(j)}^{(n)} - n\theta - d_{(j)}^{(k)} + k\theta}{n-k}\right\}$$

$$= \min_{j\in V}\max_{0\leqslant k\leqslant n-1}\left\{\frac{d_{(j)}^{(n)} - d_{(j)}^{(k)}}{n-k}\right\} - \theta.$$

因此当选取 $\theta = \lambda^*$ 时,此式就变为情形(1).

这个算法的复杂性为 $O(n\cdot m)$,其中 $m = |A|$.

3.2.2 最小平均圈的原始对偶算法

最小平均费用圈问题可以描述为如下的线性规划问题:

$$(\mathrm{P})\quad\begin{cases}\min\sum\limits_{(i,j)\in A} b_{ij}f(i,j),\\ -\sum\limits_{j\in\alpha(i)} f(i,j) + \sum\limits_{j\in\beta(i)} f(j,i) = 0, & i\in V,\\ \sum\limits_{(i,j)\in A} f(i,j) = 1,\\ f(i,j)\geqslant 0, & (i,j)\in A.\end{cases}$$

事实上,对 N 中任一回路 L,令

$$f(i,j) = \begin{cases}1/|L|, & 若(i,j)\in L,\\ 0, & 否则,\end{cases}\tag{3-7}$$

则$\{f(i,j)\}$是(P) 的可行解.

反之,若$\{f(i,j)\}$是(P) 的最优解,则$\{f(i,j)\}$是一个循环流. 事实上设

$$S = \{(i,j)\in A \mid f(i,j) > 0\},$$

则由 S 导出的有向网络 $N(S) = (V,S)$ 可由一组回路覆盖 S,设这些回路为 $C_1, C_2,\cdots, C_k$. 从而流$\{f(i,j)\}$可分解为回路上的流之并. 记 C_i 上的流为$f(i)$, $|C_i|$ 为回路 C_i 中的弧数,则

$$\sum_{(i,j)\in A} b_{ij}f(i,j) = \sum_{i=1}^{k} b(C_i)f(i) = \sum_{i=1}^{k}\frac{b(C_i)}{|C_i|}|C_i|f(i),$$

$$\sum_{(i,j)\in A} f(i,j) = \sum_{i=1}^{k}|C_i|f(i).$$

设 C_1 满足 $b(C_1)/|C_1| = \min\limits_{1\leqslant i\leqslant k}\{b(C_i)/|C_i|\}$,则

$$\sum_{(i,j)\in A} b_{ij}f(i,j) = \sum_{i=1}^{k} b(C_i)f(C_i) \geqslant \frac{b(C_1)}{|C_1|}\sum_{i=1}^{k}|C_i|f(i) = \frac{b|C_1|}{|C_1|},$$

从而由(3-7) 式定义的

$$f(i,j) = \begin{cases}1/|C_1|, & 若(i,j)\in C_1,\\ 0, & 否则\end{cases}$$

是(P) 的可行解. 这说明(P) 的最优解是最小平均费用圈.

(P) 的对偶为

$$(\mathrm{D})\ \begin{cases}\max\ \lambda, \\ \quad u_j - u_i + \lambda \leqslant b_{ij}, \quad (i,j) \in A.\end{cases}$$

设 T 是 N 的一个支撑子图,若 T 连通且恰含有一个圈 L;当 L 是有向圈时,则称 T 是一个单圈图.

容易证明,(P) 的可行基与 N 的单圈图是一一对应的.

给定 N 的一个单圈图 T,设 T 的唯一回路为 L.定义

$$f(i,j) = \begin{cases}1/|L|, & 若(i,j) \in L, \\ 0, & 其余,\end{cases}$$

则 $\{f(i,j)\}$ 为(P) 的可行解.

另一方面,在 L 上任取一点 r,把 r 作为固定点,定义

$$u_r^* = 0, \quad \lambda^* = \frac{b(L)}{|L|}.$$

若 u_i^* 已定义,则令

$$\begin{cases}u_j^* = u_i^* + b_{ij} - \lambda^*, & 若(i,j) \in T, \\ u_j^* = u_i^* - b_{ij} + \lambda^*, & 若(j,i) \in T.\end{cases} \tag{3-8}$$

因为 T 是连通的,所以对每一点 $j \in V$, u_j^* 是有定义的.

显然,若 $\{u_i^*, \lambda^*\}$ 是(D) 的可行解时,则由线性规划互补最优性准则知,L 就是最小平均费用圈.

反之,若(D) 的可行解 $\{u_i^0, \lambda^0\}$ 所确定的网络 $N(u_i^0, \lambda^0) = \{(i,j) \in A \mid u_j^0 - u_i^0 + \lambda^0 = b_{ij}\}$ 中包含有向圈,则它的任何一个有向圈都是最小平均费用圈.由此可得到

原始对偶算法:

步 0 设 $\{u_i^0, \lambda^0\}$ 为(D) 的一个可行解(开始可取 $\lambda^0 = \min\{b_{ij} \mid (i,j) \in A\}$, $u_i^0 = 0, i \in V$).

步 1 找出子网络

$$N(u_i^0, \lambda^0) = \{(i,j) \in A \mid u_j^0 - u_i^0 + \lambda^0 = b_{ij}\}.$$

若 $N(u_i^0, \lambda^0)$ 中包含有向圈 L,则算法终止,按(3-7) 式定义的 $\{f(i,j)\}$ 就是(P) 的最优解,而 $\{u_i^0, \lambda^0\}$ 为(D) 的最优解.

注意,最短路算法可用于找出 $N(u_i^0, \lambda^0)$ 中的有向圈.

若 $N(u_i^0, \lambda^0)$ 中不含有向圈,构造新的网络 $\tilde{N}(u_i^0, \lambda^0) = (\tilde{V}, \tilde{A}; d)$,其中

$$\tilde{V} = V \cup \{s\}, \quad \tilde{A} = T \cup S,$$
$$T = \{(i,j) \in A\} \mid u_j^0 - u_i^0 + \lambda^0 = b_{ij}\},$$
$$S = \{(s,j) \mid j 在 N(u_i^0, \lambda^0) 中无入弧\}.$$

若 $(i,j) \in T$,取 $d_{ij} = -1$;若 $(s,j) \in S$,取 $d_{sj} = 0$.

步 2 在 $\tilde{N}$ 中求自 s 到各点的最短路长度.设 u_j^* 表示 s 到 j 的最短路长度.那么显然对一切 $(i,j) \in T$ 有

$$u_j^* - u_i^* \leqslant -1, \quad (i,j) \in T.$$

步3　改进(D)的可行解$\{u_i^0, \lambda^0\}$.

(3-1)　若对一切$(i,j) \in A$有$u_j^* - u_i^* + 1 \leqslant 0$,则(P)无可行解.

(3-2)　计算

$$\theta = \min\left\{\frac{b_{ij} - u_j^0 + u_i^0 - \lambda^0}{u_j^* - u_i^* + 1} \mid u_j^* - u_i^* + 1 > 0, \quad (i,j) \in A\right\},$$

置

$$u_i^0 \leftarrow u_i^0 + \theta u_i^*, \quad \lambda^0 \leftarrow \lambda^0 + \theta.$$

返回步1.

这个算法的复杂性最多为$O(n^4)$.

3.2.3　最小费用循环流的最小平均圈算法

步0　求网络$N = (V, A; b, l, c)$的一个可行循环流$\{f(i,j)\}$,开始可由2.3节中的方法求出一个循环流.

步1　设$\{f(i,j)\}$是N上的可行循环流,构造N关于f的剩余网络$N(f) = (V, \tilde{A}, \tilde{b}, \tilde{l}, \tilde{c})$.其方法见式(3-5).

步2　利用3.2.1或3.2.2小节中的方法求$N(f)$中的最小平均费用圈,设它为L.

若$\tilde{b}(L) \geqslant 0$,则算法终止,$\{f(i,j)\}$已是N的最小费用的循环流.

步3　设$\tilde{b}(L) < 0$,令

$$\theta = \min\{\tilde{c}_{ij} \mid (i,j) \in L\},$$

$$f(i,j) \leftarrow \begin{cases} f(i,j) + \theta, & (i,j) \in A^+ \cap L, \\ f(i,j) - \theta, & (j,i) \in A^- \cap L, \\ f(i,j), & \text{其余}. \end{cases}$$

返回步1.

可以证明,从任一个可行循环流开始,这个算法最多迭代$O(m^2 n \mathrm{lb} n)$次必得到最小费用循环流,而每次迭代的主要计算量在于求最小平均费用圈,从而该算法的计算复杂性为$O(m^3 n^2 \mathrm{lb} n)$,故它是一个强多项式算法.

3.3　最小费用流的最短路算法

本节讨论模型(3-3),并取下界函数$l = 0$.即如下模型:

$$\begin{cases} \min \sum_{(i,j) \in A} b_{ij} f(i,j), \\ -\sum_{j \in \alpha(i)} f(i,j) + \sum_{j \in \beta(i)} f(j,i) = \begin{cases} -v, & i = s, \\ 0, & i \in V \setminus \{s,t\}, \\ v, & i = t, \end{cases} \\ 0 \leqslant f(i,j) \leqslant c_{ij}, (i,j) \in A, \end{cases} \tag{3-9}$$

其中v为一个参数,它表示流的值,这个模型对应的网络记为$N = (V, A; b, c)$,其

中 s、$t \in V$ 为 N 的发点与收点. 不妨假设 $b \geqslant 0$(为简单起见).

下面给出的算法是把 v 视为变量,即求使 v 达到最大的最小费用流. 当 v 的值给定时,这个方法也可用,只要把它视为容量即可.

相继最短路算法:

步 0　取 $f = 0$. 显然 0 流也是可行流. 由于 $b_{ij} \geqslant 0$, 所以它也是在 $v = 0$ 时的最优流,即最小费用流.

步 1　构造 N 关于可行流 f 的剩余网络 $N(f)$(见(3-5)式).

步 2　求 $N(f)$ 中自 s 到 t 的最短路或最小费用的路(这儿弧的费用为 $\tilde{b}_{ij}$):
若 $N(f)$ 中不存在 s 到 t 的路,则算法终止,$\{f(i,j)\}$ 即为 N 中最小费用的最大流.
设 P 为 $N(f)$ 中自 s 到 t 的最短路. 取

$$\theta = \min\{ \tilde{c}_{ij} \mid (i,j) \in P \}.$$

步 3　改善流 f. 置

$$f(i,j) = \begin{cases} f(i,j) + \theta, & 若 (i,j) \in P \cap A^+, \\ f(i,j) - \theta, & 若 (j,i) \in P \cap A^-, \\ f(i,j), & 其余. \end{cases}$$

返回步 1.

这个算法每次是沿最小费用的路自 s 到 t 运送 θ 个单位,因此,当第 k 次迭代后的流值为 δ,那么第 $k+1$ 次后的流值为 $\delta + \theta$,而相应的流 $\{f(i,j)\}$ 就是在 $v = \delta + \theta$ 时的最小费用流. 因此,这个算法的优点是,每次迭代后的流总是最小费用流;然而它的弱点也比较明显,即它的迭代次数与网络中的容量有关,故在一般情况下,它不是多项式的.

3.4 最小费用流的网络单纯形算法

本节讨论模型(3-1),为简单起见仍设 $l = 0$. 即

$$\begin{cases} \min \sum\limits_{(i,j)\in A} b_{ij} f(i,j), \\ -\sum\limits_{j\in\alpha(i)} f(i,j) + \sum\limits_{j\in\beta(i)} f(j,i) = a(i), \quad i \in V, \\ 0 \leqslant f(i,j) \leqslant c_{ij}, \quad (i,j) \in A. \end{cases} \tag{3-10}$$

设模型(3-10)对应的网络为 $N = (V, A; b, c)$,而 $\{f(i,j)\}$ 是模型(3-10)的一个可行流,那么 N 中的弧集 A 被流 f 划分为三部分:

$$T = \{(i,j) \mid 0 < f(i,j) < c_{ij}, (i,j) \in A\},$$
$$L = \{(i,j) \mid f(i,j) = 0, (i,j) \in A\},$$
$$U = \{(i,j) \mid f(i,j) = c_{ij}, (i,j) \in A\}.$$

若 T 中弧不含圈(不必有向圈),则称可行流 f 为极流. 若弧集 T 导出的 N 的子网络 (V, T) 连通且不含圈,则称 (V, T) 为 N 的支撑树(关于支撑树的详细定义及性质见下一章). 若 (V, T) 是 N 的支撑树,则称极流 f 是非退化的,称 T 中弧为自由

弧,而 L 与 U 中弧为限制弧.

3.4.1 某些基本性质

(1) 模型(3-10) 的最优解必能在极流中找到.设 f 是模型(3-10) 的最优解,并且使自由弧的个数最小,则此时 f 对应的 T 中不含圈.假如不然,设 W 是 T 里的一个圈,那么在 N 关于 f 的剩余网络 $N(f)$ 中,W 就对应两个方向相反的有向圈 W_1 和 W_2.由于 f 是最优的,从而 $\tilde{b}(W_1) \geq 0$, $\tilde{b}(W_2) \geq 0$.由于 $\tilde{b}(W_1) = -\tilde{b}(W_2)$,故 $\tilde{b}(W_1) = \tilde{b}(W_2) = 0$.取 $\theta_1 = \min\{\tilde{c}_{ij} \mid (i,j) \in W_1\}$, $\theta_2 = \min\{\tilde{c}_{ij} \mid (i,j) \in W_2\}$ 及 $\theta = \min\{\theta_1, \theta_2\}$.设 $\theta = \theta_k$,定义

$$f'(i,j) = \begin{cases} f(i,j) - \theta_k, & (i,j) \in W \cap W_k, \\ f(i,j) + \theta_k, & (i,j) \in W \setminus W_k, \\ f(i,j), & \text{其余}. \end{cases}$$

不难验证 f' 也是(3-10) 的可行循环流,并且其目标值与 f 的目标值相同,即 f' 也是最优解.但它对应的 T' 中的自由弧至少比 T 中的弧数少1.这与 f 的假设矛盾.

(2) 设 f^* 是一循环流,则 f^* 是最小费用的循环流,当且仅当存在位势 π^*,满足

1° 若 $(i,j) \in T$,则 $\pi_j^* - \pi_i^* = b_{ij}$;

2° 若 $(i,j) \in L$,则 $\pi_j^* - \pi_i^* \leq b_{ij}$;

3° 若 $(i,j) \in U$,则 $\pi_j^* - \pi_i^* \geq b_{ij}$.

这儿的 T、L 和 U 是由 f^* 确定的弧集 A 的划分.

模型(3-10) 的对偶是

$$\begin{cases} \max \sum_{i \in V} a(i)\pi_i - \sum_{(i,j) \in A} c_{ij}u_{ij}, \\ \pi_j - \pi_i - u_{ij} \leq b_{ij}, \\ u_{ij} \geq 0. \end{cases} \tag{3-11}$$

由线性规划的互补最优性准则,很容易证明上述结论.

一个图(有向或无向),若它连通且不含圈,则称它为树.一棵树 T,若 i 和 j 是 T 中两个顶点,并且 $[i,j]$ 边不在 T 里,那么 $T \cup [i,j]$ 有唯一一个圈.

一个网络 N,如果 N 的一个子网络是一棵树且它包含 N 的所有顶点,则称它为 N 的**支撑树**.(关于树的若干性质见下一章).

不失一般性,设 $N = (V, A; b, c)$ 连通且对任意弧 $(i,j) \in A$ 有 $b_{ij} \geq 0$ 及 $c_{ij} > 0$.由前述的基本性质(2) 可得模型(3-10) 的单纯形算法——支撑树迭代法.

3.4.2 支撑树迭代法

步0 求网络 $N = (V, A; b, c)$ 的一个可行的极流,设它为 $\{f(i,j)\}$.定义

$$T^0 = \{(i,j) \in A \mid 0 < f(i,j) < c_{ij}\}.$$

求出 N 的一个支撑树 T,使得 $T \supseteq T^0$,并定义

$$L = \{(i,j) \mid (i,j) \in A \setminus T \text{ 且 } f(i,j) = 0\},$$
$$U = \{(i,j) \mid (i,j) \in A \setminus T \text{ 且 } f(i,j) = c_{ij}\},$$

称(T,L,U)是N的一个可行划分.

步1 设(T,L,U)是N的一个可行划分,定义N的顶点位势如下:

任取一点,例如顶点$1\in V$,求解方程组

$$\begin{cases}\pi_j-\pi_i=b_{ij}, & (i,j)\in T,\\ \pi_1=0.\end{cases} \tag{3-12}$$

易证它有唯一解,设它为$\{\pi_i^*\}$.

步2 最优性检验:

(2-1) 若对一切弧$(i,j)\in L$有$\pi_j^*-\pi_i^*\leqslant b_{ij}$,进行(2-2)步;否则取一弧$(k,l)\in L$,使$\pi_l^*-\pi_k^*>b_{kl}$,进行步3,并置$L\leftarrow L-(k,l)$.

(2-2) 若对一切弧$(i,j)\in U$有$\pi_j^*-\pi_i^*\geqslant b_{ij}$,则算法结束,$\{f(i,j)\}$为最小费用流,即模型(3-10)的最优解.否则选取一弧$(k,l)\in U$,使$\pi_l^*-\pi_k^*<b_{kl}$.置$U\leftarrow U-(k,l)$.

步3 找出$T\cup\{(k,l)\}$的圈W,给W一个方向,使W的方向与(k,l)方向一致.这样W中的弧可分为两部分:

$$W^+=\{(i,j)\in W\mid (i,j)\text{的方向与}W\text{方向一致}\},$$
$$W^-=\{(i,j)\in W\mid (i,j)\text{的方向与}W\text{方向相反}\}.$$

定义

$$\delta_{ij}=\begin{cases}c_{ij}-f(i,j), & \text{若}(i,j)\in W^+,\\ f(i,j), & \text{若}(i,j)\in W^-,\end{cases}$$
$$\theta=\min\{\delta_{ij}\mid (i,j)\in W\}=\delta_{pq}.$$

改进流f如下:

置

$$f(i,j)\leftarrow\begin{cases}f(i,j)+\theta, & \text{若}(i,j)\in W^+,\\ f(i,j)-\theta, & \text{若}(i,j)\in W^-,\\ f(i,j), & \text{其余}.\end{cases}$$

置 $$T\leftarrow T+(k,l)-(p,q);$$

若$(p,q)\in W^+$,置$U\leftarrow U+(p,q)$;若$(p,q)\in W^-$,置$L\leftarrow L+(p,q)$.则(T,L,U)是N的一个可行划分.返回步1.

注意:为了保证算法的有限性,可以采用单纯形算法的Bland则法:设N中的弧编号为$1,2,\cdots,m$.在第2步的最优性检查中,按弧的编号由小到大检查,找出不满足要求的最小标号的弧作为(k,l).同时在(p,q)选择中,在所有的$\delta_{ij}=\theta$中选取标号最小的弧作为(p,q).

3.5 最小费用流的OK算法

本节讨论有向网络$N=(V,A;b,l,c)$上的最小费用流问题,它可以是模型(3-1)、(3-2)和(3-3)中任何一种.为简单起见,假设对任意的$(i,j)\in A$,$c_{ij}<\infty$并且$l_{ij}<c_{ij}$.

假设已有N上的可行流$\{f(i,j)\}$及对应的位势$\{\pi_i\}$(它可以由上节中算法的

第1步求出),定义弧集:

$$\alpha = \{(i,j) \in A \mid \pi_j - \pi_i < b_{ij} \text{且} f(i,j) = l_{ij}\},$$
$$\beta = \{(i,j) \in A \mid \pi_j - \pi_i = b_{ij} \text{且} l_{ij} \leqslant f(i,j) \leqslant c_{ij}\},$$
$$\gamma = \{(i,j) \in A \mid \pi_j - \pi_i > b_{ij} \text{且} f(i,j) = c_{ij}\},$$
$$a = \{(i,j) \in A \mid \pi_j - \pi_i < b_{ij} \text{且} f(i,j) > l_{ij}\},$$
$$b = \{(i,j) \in A \mid \pi_j - \pi_i > b_{ij} \text{且} f(i,j) < c_{ij}\}.$$

显然由3.4.1节中的(2)知,当 $a \cup b = \varnothing$ 时,$\{f(i,j)\}$ 即为最小费用流.把 $\alpha \cup \beta \cup \gamma$ 中弧称为良弧,而 $a \cup b$ 中的弧称为劣弧.对每一弧 $(i,j) \in A$,定义一个**劣度** $K(i,j)$

$$K(i,j) = \begin{cases} 0, & \text{若}(i,j) \in \alpha \cup \beta \cup \gamma, \\ f(i,j) - l_{ij}, & \text{若}(i,j) \in a, \\ c_{ij} - f(i,j), & \text{若}(i,j) \in b. \end{cases}$$

流 f 的劣度定义为 $K(f) = \sum_{(i,j)\in A} K(i,j)$.

OK算法是在保证良弧永远是良弧的情况下,用调整可行流 f 或改变位势 π,使 $K(f)$ 单调下降,直到某一个 f,使 $K(f) = 0$ 为止,即得到最小费用流.

为了叙述方便,把 β 中的弧再细分:

$$\beta_1 = \{(i,j) \in \beta \mid f(i,j) = l_{ij}\},$$
$$\beta_2 = \{(i,j) \in \beta \mid f(i,j) = c_{ij}\},$$
$$\beta_0 = \{(i,j) \in \beta \mid l_{ij} < f(i,j) < c_{ij}\}.$$

为了保证良弧始终是良弧,而劣弧的劣度下降,那么 f 和 π 的改变必须遵循下述规则:

(1) 对 $(i,j) \in a$,$f(i,j)$ 只可能减小,并且最多减至 l_{ij};$\pi_j - \pi_i$ 只能增加,但最多增至 b_{ij};

(2) 对 $(i,j) \in b$,$f(i,j)$ 只可能增加,但最多增至 c_{ij};而 $\pi_j - \pi_i$ 只能下降,但最多减至 b_{ij}.

(3) 对 $(i,j) \in \beta$,$f(i,j)$ 允许在 l_{ij} 和 c_{ij} 之间变化.当 $(i,j) \in \beta_0$ 时,$\pi_j - \pi_i$ 不能改变;当 $(i,j) \in \beta_1$ 时,$\pi_j - \pi_i$ 可任意减小;当 $(i,j) \in \beta_2$ 时,$\pi_j - \pi_i$ 可任意增加.

(4) 当 $(i,j) \in \alpha$ 时,$f(i,j)$ 不能改变;而 $\pi_j - \pi_i$ 可以任意减小,亦可以增加,但最多增至 b_{ij}.

(5) 当 $(i,j) \in \gamma$ 时,$f(i,j)$ 不能改变;而 $\pi_j - \pi_i$ 可以任意增大,也可以减小,不过至多减至 b_{ij}.

按这些规则来改进可行流 f 和位势 π,就可得到如下算法:

OK算法:

步0　设 f 是 N 上的可行流向量,π 是 N 上的位势(开始可利用第2章流的算法求 f;π 可由3.4节中树迭代算法的第1步确定,也可以任意给定).

步1　由可行流 f 和位势 π,确定 N 的弧集 A 的划分 $(\alpha,\beta,\gamma,a,b)$ 及 $\beta = (\beta_0, \beta_1, \beta_2)$.

步2　若 $a \cup b = \varnothing$,则 f 为最小费用流,算法终止.否则,任取一弧 $(p,q) \in$

$a \cup b$,如果$(p,q) \in a$,则置 $\bar{s} \leftarrow p, \bar{t} \leftarrow q, \delta \leftarrow f(p,q) - l_{pq}$;若$(p,q) \in b$,则置 $\bar{s} \leftarrow q, \bar{t} \leftarrow p, \delta \leftarrow c_{pq} - f(p,q)$,转步 3.

步 3　求自 $\bar{s}$ 到 $\bar{t}$ 的最短(含弧数最小)链 P,并且不含弧$(\bar{s},\bar{t})$及$(\bar{t},\bar{s})$.

(3-1)　给 $\bar{s}$ 标以($\varnothing$),且未检查.

(3-2)　按先标号先检查的原则,对每一个有标号且未检查的点 i,进行如下检查:

(3-2-1)　对一切未标号的点 j,若弧$(i,j) \in b \cup \beta_0 \cup \beta_1$,则给 j 标以$(+i)$;若弧$(j,i) \in a \cup \beta_0 \cup \beta_2$,则给 j 标以$(-i)$.这样 j 变为有标号且未检查的点;而 i 变为已检查过的点.若 $\bar{t}$ 得到标号,则进行步 4.

(3-2-2)　若所有标号点都已检查过,没有得到新的标号点,并且 $\bar{t}$ 未标号,则转步 5.

步 4　沿顶点的标号返回追踪,找出 $\bar{s}$ 到 $\bar{t}$ 的链 P.记 P^+ 为 P 上的顺向弧(与 P 的走向一致的弧),P^- 为 P 上的反向弧,定义

$$\theta_1 = \min\{c_{ij} - f(i,j) \mid (i,j) \in P^+\},$$
$$\theta_2 = \min\{f(i,j) - l_{ij} \mid (i,j) \in P^-\},$$
$$\theta = \min\{\theta_1, \theta_2, \delta\},$$

令

$$f(i,j) \leftarrow \begin{cases} f(i,j) + \theta, & 若(i,j) \in P^+, \\ f(i,j) - \theta, & 若(i,j) \in P^-, \\ f(i,j), & 若(i,j) \in A - P. \end{cases}$$

进而若$(p,q) \in a$,置$f(p,q) \leftarrow f(p,q) - \theta$;若$(p,q) \in b$,则置$f(p,q) \leftarrow f(p,q) + \theta$.从新的可行解 f 返回步 1.

步 5　设所有标号顶点的集合为 X,而未标号的点的集合为 $\bar{X}$,则 $\bar{s} \in X, \bar{t} \in \bar{X}$,从而$(X,\bar{X})$是 $\bar{s}$-$\bar{t}$ 割.显然,$(X,\bar{X}) \subseteq \alpha \cup \gamma \cup a \cup \beta_2$,并且$(\bar{X},X) \subseteq b \cup \gamma \cup \alpha \cup \beta_1$.定义

$$\delta_1 = \min\{b_{ij} - \pi_j + \pi_i \mid (i,j) \in (X,\bar{X}) \cap \{\alpha \cup a\}\},$$

$$\delta_2 = \min\{\pi_j - \pi_i - b_{ij} \mid (i,j) \in (\bar{X},X) \cap \{b \cup \gamma\}\},$$
$$\epsilon = \min\{\delta_1, \delta_2\},$$

置

$$\pi_i \leftarrow \begin{cases} \pi_i + \epsilon, & 若 i \in \bar{X}, \\ \pi_i, & 若 i \in X. \end{cases}$$

然后用新的位势 π 返回步 1.

这个算法每次迭代后,或者改进了可行流或者改善了位势.从而可以保证它的收敛性.

4　最优树与树形图

4.1　树的基本概念

一个图 $G=(V,E)$，若 G 连通且无圈，则称它为一棵树. 设 $T=(V,E)$ 是 n 个顶点的图，则下述结论等价：

(1) T 是一棵树.

(2) T 有 $n-1$ 条边且无圈.

(3) T 有 $n-1$ 条边且连通.

(4) T 连通，但若从 T 中去掉任一条边，则图就不连通了.

(5) T 中任何一对点之间有唯一一条链.

(6) T 无圈，若 T 中任一对不相邻的点之间连一条边，则有唯一一个圈.

以上是关于树的一些等价定义，其证明是简单的.

设 $G=(V,E)$ 是一个简单图，即每两个点之间最多有一条边，并且没有两个端点相重合的边. 若 G 有一个子图 $T=(U,E')$ 使 $U=V$，$E'\subseteq E$，则称 T 为 G 的支撑子图；若 G 的支撑子图 T 是一棵树，则称 T 是 G 的支撑树. 设 T 是 G 的支撑树，那么从 T 中去掉任何一条边 $[p,q]$，则由性质(4)知，T 被分为两部分 T_p 和 T_q，其中 p 在 T_p 中，而 q 在 T_q 中. 记 G 中一个端点在 T_p 中，而另一个端点在 T_q 中的边之集合为 $\Omega(p,q)$，则称 $\Omega(p,q)$ 是关于树 T 的一个**基本割**. 由于 G 有 n 个点而 T 有 $n-1$ 条边，所以 G 有 $n-1$ 个基本割.

另一方面，对 G 中每一条不在 T 中的边 $[p,q]$，将 $[p,q]$ 加到 T 中，则由性质(6)知，$T+[p,q]$ 有唯一一个圈，记它为 $C(p,q)$，并称它为关于 T 的**基本圈**. 显然，若 G 有 m 条边，那么它有 $m-n+1$ 个基本圈.

设 T_1 和 T_2 是 G 的两棵支撑树，定义 T_1 和 T_2 之间的距离为 $|T_1-T_2|$，记为 $d(T_1,T_2)=|T_1-T_2|$.

设 $\{e_1,e_2,\cdots,e_{n-1}\}$ 是 T_1 中边的任一个顺序，那么 T_2 中边必存在一个顺序，记它为 $\{a_1,a_2,\cdots,a_{n-1}\}$，使得对任意 $i=1,2,\cdots,n-1$，$T_1-\{e_1,e_2,\cdots,e_i\}+\{a_1,a_2,\cdots,a_i\}$ 也是 G 的一棵支撑树.

4.2　最优支撑树

设 $N=(V,E;w)$ 是一个无向的连通网络，每一边 $e\in E$ 有一个权 $w(e)$. N 的一棵支撑树 T 的权定义为 T 中边的权之和，即 $w(T)=\sum_{e\in T}w(e)$. 权最小的支撑树，称为 G 的**最小权支撑树**或**最小树**.

最小树有如下的性质：

(1) 设 T 是连通网络 $N=(V,E;w)$ 的一棵支撑树，则 T 是最小权支撑树当且仅当对每一条边 $e\in E-T$，有 $w(e)=\max\{w(e')\mid e'\in C(e)\}$，其中 $C(e)$ 为 $T+e$ 中的圈.

(2) 设 T 是连通网络 $N=(V,E;w)$ 的一棵支撑树，则 T 是最小权支撑树当且仅当对每一条边 $e\in T$，有 $w(e)=\min\{w(e')\mid e'\in \Omega(e)\}$，其中 $\Omega(e)$ 为 G 关于 T 的基本割.

4.2.1　线性规划模型与贪婪(greedy)算法

设 $N=(V,E;w)$ 为一连通网络，对 $S\subseteq E$，用 $r(S)$ 表示由 S 导出的子图中最大森林的边数，即

$r(S)=\max\{|F|\mid F\subseteq S$ 且 F 中的边不构成圈$\}$.

N 的最小权支撑树问题可描述为

$$\begin{cases}\min\sum_{e\in E}w(e)x_e,\\ \sum_{e\in E}x_e=r(E),\\ \sum_{e\in[S]}x_e\leqslant r(S), & S\subset E,\\ x_e\geqslant 0, & e\in E,\end{cases}\tag{4-1}$$

其中 $[S]=\{e\in E\mid r(S+e)=r(S)\}$，称 $[S]$ 为 S 的闭包.

显然，对 N 的任一支撑树 T，若令

$$x_e=\begin{cases}1, & 若\ e\in T,\\ 0, & 其余,\end{cases}$$

则 $\{x_e\}$ 是问题(4-1)的一个可行解. 若问题(4-1)的最优解为 0-1 解，则 $T=\{e\mid x_e=1\}$，就是 N 的支撑树，并且必是最小权的. 亦即

$$\begin{cases}\min\sum_{e\in E}w(e)x_e,\\ \sum_{e\in E}x_e=r(E),\\ \sum_{e\in[S]}x_e\leqslant r(S), & S\subset E,\\ x_e=0,1, & e\in E\end{cases}\tag{4-2}$$

的最优解为 N 的最小权支撑树. 然而有趣的是，(4-2)式的最优解也必是(4-1)式的最优解，而(4-1)式的基本可行解也是(4-2)式的可行解，这样一来，求 N 的最小树可以通过解线性规划问题(4-1)得到. 然而，由于问题(4-1)中约束条件太多，不太可能用单纯形算法求解，这儿用贪婪算法求解(4-1)式.

贪婪算法：

步 0　将 N 中边按权由小到大排列，不妨设 $w(e_1)\leqslant w(e_2)\leqslant\cdots\leqslant w(e_m)$. 置 $T\leftarrow\varnothing$，$i\leftarrow 1$.

步 1　若 $i > m$,算法终止,T 为所求.

步 2　若 $T + e_i$ 不含圈,则置 $T \leftarrow T + e_i$;若 $T + e_i$ 包含圈,则 T 不变;置 $i \leftarrow i + 1$ 返回步 1.

设算法结束时得到的边集为 T,定义

$$x_e^* = \begin{cases} 1, & 若\ e \in T, \\ 0, & 若\ e \in E - T, \end{cases}$$

则显然 $\{x_e^*\}$ 是(4-1)式的可行解.下面用线性规划的互补最优性条件证明 $\{x_e^*\}$ 是(4-1)式的最优解,即 T 为 N 的最小树.为此考虑(4-1)式的对偶:

$$\begin{cases} \max \sum_{S \subset E} - r(S) y_S + r(E)\lambda \\ \lambda - \sum_{[S] \ni e} y_S \leqslant w(e), & e \in E, \\ y_S \geqslant 0, & S \subset E. \end{cases} \tag{4-3}$$

不失一般性,设算法所选取 T 中的边依次为 $e_1, e_2, \cdots, e_k$,其中 $k = r(E)$.记 $S_i = \{e_1, e_2, \cdots, e_i\}, i = 1, \cdots, k$;定义 $\lambda^* = w(e_k), y_{S_k}^* = 0, y_{S_{k-1}}^* = w(e_k) - w(e_{k-1}), \cdots, y_{S_1}^* = w(e_2) - w(e_1)$,其余 $y_S^* = 0$.显然 $y_S^* \geqslant 0$.

首先验证这样定义的 $\{\lambda^*, y_{S_i}^*\}$ 是(4-3)式的可行解.

事实上,对任意的 $e \in E$,必存在某个 i,使 $e \in [S_i]$,而对一切 $j < i, e \notin [S_j]$.由 S_p 和 $[S_p]$ 的定义知,对一切 $q \geqslant p$,有 $[S_p] \subset [S_q]$.因此

$$\begin{aligned} \lambda^* - \sum_{[S] \ni e} y_S^* &= \lambda^* - \sum_{j=i}^{k} y_{S_j}^* = \lambda^* + w(e_i) - w(e_k) \\ &= W(e_i) \leqslant W(e). \end{aligned}$$

这儿 $w(e_i) \leqslant w(e)$ 是由贪婪算法而得.从而证明了 $\{y_S^*, \lambda^*\}$ 是(4-3)式的可行解.

下证 $\{x_e^*\}$ 和 $\{\lambda^*, y_S^*\}$ 满足线性规划的互补最优性准则,即

1° $x_e^* > 0 \ \Rightarrow\ \lambda^* - \sum_{[S] \ni e} y_S^* = w(e)$;

2° $y_S^* > 0 \ \Rightarrow\ \sum_{e \in [S]} x_e^* = r(S)$.

事实上,由 x_e^* 的定义知,只有 $e \in T$,才有 $x_e^* = 1 > 0$.对 $i = 1, 2, \cdots, k, e_i \in [S_i]$,而对 $j < i$,则 $e_i \notin [S_j]$.所以,对 $x_{e_i}^* > 0$,有

$$\lambda^* - \sum_{[S] \ni e_i} y_{S_i}^* = \lambda^* - \sum_{j=i}^{k} y_{S_j}^* = w(e_i).$$

另一方面,由 y_S^* 的定义知,只有 $y_{S_i}^*$ 有可能大于 0, $i = 1, 2, \cdots, k$.

若 $y_{S_i}^* > 0$,由 $[S_i]$ 的定义知,$x_{e_j}^* = 1, j = 1, 2, \cdots, i$,而对一切 $e \in [S_i] - S_i$ 有

$x_e^* = 0$. 同时 $r(S_i) = r([S_i])$. 从而 $\sum_{e \in [S_i]} x_e^* = \sum_{j=1}^{i} x_{e_j}^* = i = r(S_i)$.

因此，由线性规划的最优性准则知，$\{x_e^*\}$ 为(4-1)式的最优解，即 T 为最小权支撑树.

贪婪算法的计算复杂性，主要是第0步中把 m 条边按权由小到大排队. 把 m 条边由小到大排队的计算复杂性为 $O(m\text{lb}m)$，这也是贪婪算法的复杂性上界.

4.2.2　最小树的普锐姆算法

最小树的普锐姆(Prim)算法是基于本节中最小树的性质2°. 其思想是每次找割集中的一条最短边，这样 $n-1$ 个割集中的 $n-1$ 条边就构成一棵最小支撑树. 为方便起见，把 N 中诸边用其两个端点表示，如 $e = [i,j]$，相应地，边 e 的权 $w(e)$ 记为 w_{ij}. 另外，若 $[i,j]$ 不是 N 中的边，则记 $w_{ij} = \infty$. 最后记 $V = \{1,2,\cdots,n\}$.

普锐姆算法：

步0　置 $S = \{1\}$，$T = \{2,3,\cdots,n\}$，$\boldsymbol{R} = (r_2, r_3, \cdots, r_n)$，$u_j = w_{1j}$，$r_j = 1$，$j = 2,3,\cdots,n$.

步1　在 T 中找一点 k，使

$$u_k = \min_{j \in T}\{u_j\}.$$

置 $S \leftarrow S \cup \{k\}$，$T \leftarrow T - \{k\}$.

若 $T = \varnothing$，算法终止，$\{[2,r_2],[3,r_3],\cdots,[n,r_n]\}$ 构成 N 的最小权支撑树.

步2　对 T 中每一点 j，修正 u_j 如下：

若 $u_j > w_{kj}$，则置 $u_j \leftarrow w_{kj}$，$r_j \leftarrow k$；

若 $u_j \leqslant w_{kj}$，则 u_j 不变.

返回步1.

普锐姆算法的正确性，可由4.2节中最小树的性质直接推出. 这个算法的复杂性在于第1步. 若 T 中有 k 个点，则从 k 个数中找一个最小的需做 $k-1$ 次比较，而第1步要进行 $n-1$ 次. 故第1步总的运算次数为 $\sum_{i=2}^{n-1}(n-i) = (n-1)(n-2) \approx O(n^2)$，所以普锐姆的算法复杂性为 $O(n^2)$.

4.2.3　最小树的破圈法

破圈法基于本节最小树的性质1°. 每次去掉一个圈上的最长边，直到剩下的图不含圈为止. 此时剩余图必为最小树.

破圈法：

步0　$N = (V,E;w)$ 是连通图，置 $T \leftarrow E$.

步1　若 T 中的边不构成圈，算法终止. T 为 N 的最小权支撑树.

步2　找出 T 中任一个圈 L 及 L 上的最大权边 e，置 $T \leftarrow T - e$. 返回步1.

此算法也可以用探测图的连通性代替寻找圈，即将 N 的边由权自大到小排列，不妨设为 $w(e_1) \geqslant w(e_2) \geqslant \cdots \geqslant w(e_m)$. 然后对 $i = 1,2,\cdots,m$ 逐一检查 e_i，若

从图中去掉 e_i 后，剩余图仍连通，则就把 e_i 去掉，对余下图重复这过程；若去掉 e_i 后不连通，则继续检查 e_{i+1}. 如此等等.

破圈算法是贪婪算法的对偶，它的计算复杂性与贪婪算法的复杂性基本相同.

4.3 第 k 最小权支撑树

设 $N=(V,E;w)$ 是一个连通网络. 记 $\mathscr{T}$ 为 N 的所有支撑树的集合，$F_1,F_2,\cdots,F_p$ 是 $\mathscr{T}$ 的一个划分，使得对任意 $i=1,2,\cdots,p$，若 T_i、$T'_i\in F_i$，则 $w(T_i)=w(T'_i)$，并且当 $i\neq j$，$T_i\in F_i$，$T_j\in F_j$ 时，则 $w(T_i)\neq w(T_j)$，其中 $w(T_i)=\sum_{e\in T}w(e)$ 称为 T_i 的权，即 $F_1,F_2,\cdots,F_p$ 是按树的权大小对 $\mathscr{T}$ 的一个划分. 不妨设当 $i<j$ 时，$T\in F_i$，$T'\in F_j$，则有 $w(T)<w(T')$. 因此 F_1 是 N 的所有最小权支撑树的集合；而 F_p 是 N 的所有最大权支撑树的集合. 一般地，F_k 是 N 的第 k 最小支撑树集合. 用 w_i 表示 F_i 中树的权，则有

$$w_1<w_2<\cdots<w_p.$$

对任意一棵树 $T\in\mathscr{T}$ 和非负整数 k，定义

$$L_k(T)=\{T'\in\mathscr{T}\mid d(T,T')\leqslant k\}.$$

显然有

1° $T=L_0(T)\subseteq L_1(T)\subseteq\cdots\subseteq L_{n-1}(T)=\mathscr{T}$，其中 n 为网络 N 的顶点数；

2° 对任意 k，$T'\in L_k(T)$ 当且仅当 $T\in L_k(T')$.

目的是希望找到一个算法，对任意的 $k>0$，可求出 N 的一个第 k 最小支撑树 T，即求出 F_k 中一棵树.

前一节所叙述的算法都可用于求 F_1 或 F_p 中的树，而对一般的 $k>0$，还没有发现好算法. 但是可以证明：对任意 $T\in F_1$ 和任意的 k，则有

$$F_k\cap L_{k-1}(T)\neq\varnothing.$$

亦即，从任一棵最小权支撑树出发，最多经过 $k-1$ 条边的交换就可得到第 k 最小权支撑树(见文献[3].

例如 $k=2$ 时，T 是最小权支撑树. 对 $E-T$ 中每一条边 e_i，考察 $T+e_i$ 中的圈 $C(e_i)$，找出满足

$$w(e_i)-w(e^i)=\min\{w(e_i)-w(e')\mid e'\in C(e_i)\text{ 且 }w(e')\neq w(e_i)\}$$

的边 e^i，然后求

$$\min\{w(e_i)-w(e^i)\mid e_i\in E-T\}=w(e_0)-w(e^0),$$

则
$$T^0=T+e_0-e^0\in F_2.$$

仿此也可考虑 $k=3$ 的情况，但对一般的 k 就比较复杂了.

4.4 最小比例树与最均匀树

设 $N=(V,E;b,w)$，对每一边 e 有两个权 $b(e)$ 和 $w(e)$，它们分别表示建设路 e 的费用和收益. N 的一棵支撑树 T 的效益定义为

$$b(T)/w(T) = \sum_{e\in T} b(e) \Big/ \sum_{e\in T} w(e).$$

目的是求 N 的一棵支撑树 T，使 $b(T)/w(T)$ 最小(这儿假定对任一边 $e \in E$，有 $w(e) > 0, b(e) \geq 0$)．这就是所谓的**最小比例树问题**．

对 N 的任一棵支撑树 T，记 $\lambda(T) = b(T)/w(T)'$．

设 T 是 N 的一棵支撑树，则 T 是 N 的最小比例树，当且仅当存在 $\lambda > 0$，使得 $b(T) = \lambda w(T)$ 并且对 N 的任何支撑树 T' 有 $b(T') - \lambda w(T') \geq 0$．

根据这一性质可建立求最小比例树的如下算法：

步 0　任取 N 的一棵树 T，置 $\lambda \leftarrow b(T)/w(T)$，$F \leftarrow T$．

步 1　计算 $c_\lambda(e) \leftarrow b(e) - \lambda w(e), e \in E$．

步 2　以 $C_\lambda(e)$ 为网络 N 中边 e 的权，求 N 的最小枝支撑树 T．

若 $C_\lambda(T) = 0$ 或 $T = F$，则算法终止．T 为 N 的最小比例树；否则，置 $\lambda \leftarrow b(T)/w(T)$，$F \leftarrow T$．返回步 1．

由于一个网络的支撑树的数目是有限的，因此算法的有限性是显然的．

设 T 是一棵树，T 的**均匀度** $b(T)$ 定义为 T 中边的最大权与最小权的差，即

$$b(T) = \max_{e\in T} w(e) - \min_{e\in T} w(e).$$

给定一个赋权图 $G = (V, E; w)$，如何求出 G 的最均匀支撑树?所谓最均匀支撑树是指 $b(T)$ 最小的支撑树．

设 T 是 G 的一棵支撑树，记 T 中边的最大权与最小权分别为 $M(T)$ 和 $m(T)$，则 T 是 G 的最均匀树，当且仅当$(V, E - E^+)$ 和$(V, E - E^-)$ 都是非连通图．其中 $E^+ = \{e \in E \mid w(e) \geq M(T)\}$，$E^- = \{e \in E \mid w(e) \leq m(t)\}$．

由上可得如下的求最均匀支撑树的算法：

步 0　将 G 中边按权由小到大排列．不失一般性，设 $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$．置 $k \leftarrow 1, R \leftarrow \infty, L \leftarrow \varnothing, G_k = G$．

步 1　若 G_k 中不存在支撑树，即 G_k 不连通，算法终止．L 为所求，其均匀度为 R．否则，求出 G_k 的最小权支撑树 T_k．

步 2　计算 $b(T_k)$，若 $b(T_k) < R$，置 $R \leftarrow b(T_k), L \leftarrow T_k$．

步 3　置 $G_{k+1} = G_k - \{e_k\}, k \leftarrow k + 1$．返回步 1．

4.5　最优树形图

一个有向图 $H = (V, A)$，若 H 中每一个顶点入次最多为 1，并且若不计弧的方向时它是一个树，则称 H 为**树形图**．

树形图有如下性质：

(1) 树形图恰有一个顶点入次为 0，此点称为根，其余各点入次恰为 1．出次为 0 的点称为叶子或端点．

(2) 一个树形图，从它的根到每一点恰有一条路．

如果有向图 $H = (V, A')$ 是有向图 $N = (V, A)$ 的一个支撑子图，并且 H 是一个树形图，则称 H 为 N 的树形图．一个树形图的权定义为树形图中所有弧的权之

和.一个有向网络 $N=(V,A;w)$ 的最小权树形图或最小树形图,是指在 N 的所有树形图中具有权最小的一个树形图.

给定一个网络 $N=(V,A;w)$,可以构造另一个网络 $N'=(V',A';w')$,其中

$$V'=V\cup\{r\},\quad A'=A\cup\{(r,j)\mid j\in V\},$$

$$w'_{ij}=\begin{cases}M, & i=r,j\in V,\\ w_{ij}, & (i,j)\in A,\end{cases}$$

M 是充分大的数.

可以证明,求 N 的最小树形图等价于求 N' 中以 r 为根的最小树形图.

事实上,假如 H 是 N 的最小树形图,其根为 v_0,那么 $H'=H\cup\{(r,v_0)\}$ 是 N' 的以 r 为根的树形图,并且它们的权之间的关系为

$$W'(H')=W(H)+M.$$

由于 M 充分大且为常量,故 H' 必是 N' 的最小树形图.

反之,若 H' 是 N' 的以 r 为根的最小树形图.由于 M 充分大,则 $H=H'-\{r\}$ 必是 N 的最小树形图.事实上,H 必是树形图,否则 H' 至少含自 r 出发的两条弧,从而 $w'(H')\geqslant 2M+w(H)$.当 N 有树形图时,设 T 为 N 的树形图,v 为 T 的根,则 $T'=T\cup\{(r,v)\}$ 为 N' 的以 r 为根的树形图,从而

$$w'(T')=w(T)+M<2M+w(H)\leqslant W'(H').$$

这说明 H' 不是 N 的最小树形图.矛盾.

因此,不失一般性,下面仅叙述如何求一个网络 $N=(V,A;w)$ 的固定根的最小树形图的算法.

步 0　设 v_0 为 $N=(V,A;w)$ 中的一个指定的点,求以 v_0 为根的 N 的最小树形图.置

$$k\leftarrow 0,N_k\leftarrow N,V_k\leftarrow V,A_k\leftarrow A,w^{(k)}\leftarrow w,n_k\leftarrow|V_k|.$$

步 1　对每一点 $v_i\in V_k-\{v_0\}$,选取进入 v_i 的最小权的弧.设选取的弧集为 M_k.

若 $|M_k|<n_k-1$,算法终止,N 中不含以 v_0 为根的树形图.

若 N_k 的导出子图(V_k,M_k)不含回路(有向圈),则 M_k 是 N_k 中以 v_0 为根的最小树形图.转步 3.

步 2　找出子图(V_k,M_k)中所有回路,记它们为 $C_i^{(k)}=\{v_1{}^i,v_2{}^i,\cdots,v_{l_i}{}^i\}$,$i=1,2,\cdots,t_k$.注意 $i\neq j$ 时 $C_i^{(k)}$ 和 $C_j^{(k)}$ 是不相交的.

在 N_k 中将每一个 $C_i^{(k)}$ 收缩为一点 y_i^k,$i=1,2,\cdots,t_k$,这样由 N_k 得到一个新的网络 $N_{k+1}=(V_{k+1},A_{k+1};w^{(k+1)})$,其中 $A_{k+1}=A_{k+1}^1\cup A_{k+1}^2$,而

$$A_{k+1}^1=\{(i,j)\in A_k\mid j\notin\bigcup_{l=1}^{t_k}C_l^{(k)}\}.$$

$$A_{k+1}^2=\bigcup_{l=1}^{t_k}\{(i,j)\in A_k\mid j\in C_l^{(k)}\text{ 且 }i\notin C_l^{(k)}\},$$

$$w_{ij}^{(k+1)}=\begin{cases}w_{ij}^{(k)}, & (i,j)\in A_{k+1}^1,\\ w_{ij}^{(k)}-w_{vj}^{(k)}, & (i,j)\in A_{k+1}^2\text{ 且 }j\in C_l^{(k)},(v,j)\in C_l^{(k)}.\end{cases}$$

然后置 $k \leftarrow k+1$,返回第 1 步.

注意:可以去掉 N_{k+1} 中的环,并且若 i 到 j 有多条弧时可以只留一条长度最短的弧.

例如图 4-1,第 1 步得

$$M_0 = \{(7,1),(1,2),(9,3),(3,4),(4,5),(5,6),(8,7),(2,8),(5,9)\},$$

$$|M_1| = 9 = n_1 - 1,$$

M_1 中的回路为

$$C_1^{(0)} = \{(1,2),(2,8),(8,7),(7,1)\},$$

$$C_2^{(0)} = \{(3,4),(4,5),(5,9),(9,3)\}.$$

在 N_0 中收缩 $C_1^{(0)}$ 和 $C_2^{(0)}$,得收缩点 y_1^0 和 y_2^0,重新定义弧长,这样得 N_1,去掉环和重弧之后,N_1 为图 4-2 所示.

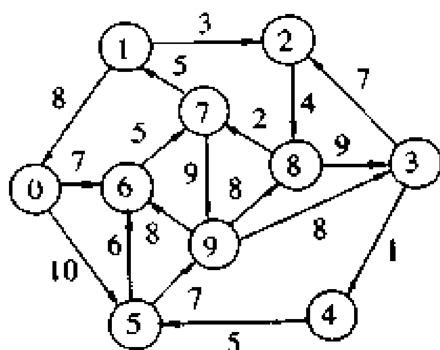


图 4-1

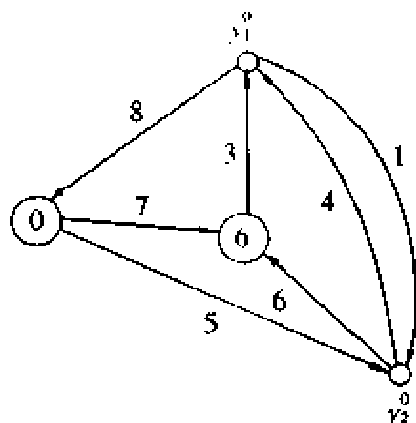


图 4-2

对 N_1 继续第 1 步,得到 N_2,如图 4-3 所示.

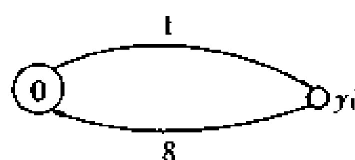


图 4-3

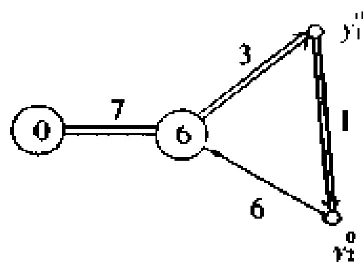


图 4-4

再对 N_2 进行第 1 步得 $M_2 = \{(0, y_1{}^1)\}$,它是 N_2 的最小树形图.

步 3　设 $H_k = M_k$ 为 N_k 的以 v_0 为根的树形图,

(3-1)　若 $k=0$,则 H_0 为 N_0 的以 v_0 为根的最小树形图,算法结束.

(3-2)　若 $k>0$,将 H_k 的收缩点还原为回路,这样得回路 $C_i^{(k-1)}$, $i=1,2,\cdots,t_{k-1}$.将 $C_i^{(k-1)}$ 中与 H_k 有公共终点的一条弧去掉,这样就得到 N_{k-1} 的最小树形图 H_{k-1}.置 $k \leftarrow k-1$,返回(3-1).

例如上例,$H_2 = M_2 = \{(0, y_1^1)\}$,把 y_1^1 还原为回路 $C_1^{(1)} = \{(y_1^0, y_2^0),(y_2^0,6),(6,y_1^0)\}$,弧$(0,y_1^1) = (0,6)$,它与 $C_1^{(1)}$ 上的弧$(y_2^0,6)$ 有公共终点,故去掉弧$(y_2^0,6)$ 得 N_1 上的树形图 $H_1 = \{(0,6),(6,y_1^0),(y_1^0,y_2^0)\}$,如图 4-4.

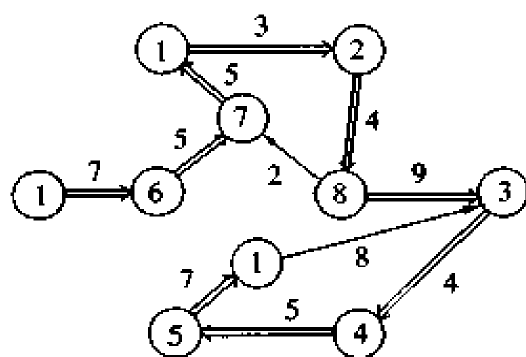


图 4-5

再将 H_1 中的收缩点 y_1^0 和 y_2^0 还原为回路 $C_1^{(0)}$、$C_2^{(0)}$，去掉 $C_1^{(0)}$ 上的弧(8,7)和 $C_2^{(0)}$ 上的弧(9,3)得图4-5.

$H_0 = H_1 \cup C_1^{(0)} \cup C_2^{(0)} - \{(8,7),(9,3)\}$ 为 N 上的以0为根的最小树形图，如图4-5中双线所示.

不难看出，若把 w 改为 $-w$，则上述算法求出的树形图就是原网络中以 v_0 为根的最大权树形图.

4.6　最优划分限制支撑树

设 $N=(V,E;w)$ 是一个无向网络，记 $\mathscr{T}=(B_1,B_2,\cdots,B_m)$ 是边集 E 的一个划分，即 $\bigcup_{i=1}^{m} B_i = E, i\neq j, B_i\cap B_j=\varnothing$. 对每一个 B_i 有两个非负整数 a_i 和 d_i，使 $a_i \leqslant d_i, i=1,2,\cdots,m$.

N 的一棵支撑树 T，若 T 满足对一切 $i=1,2,\cdots,m$，

$$a_i \leqslant |T\cap B_i| \leqslant d_i$$

则称 T 为划分限制支撑树. N 中使权最小的划分限制支撑树，称为 N 的**最小划分限制支撑树**.

设 T 是 N 的支撑树，定义 T 的划分：

$$A=\{i \mid |T\cap B_i| = a_i < d_i, i=1,2,\cdots,m\},$$
$$B=\{i \mid |T\cap B_i| = d_i > a_i, i=1,2,\cdots,m\},$$
$$C=\{i \mid |T\cap B_i| < a_i, i=1,2,\cdots,m\},$$
$$D=\{i \mid |T\cap B_i| > d_i, i=1,2,\cdots,m\},$$
$$I=\{i \mid a_i < |T\cap B_i| < d_i, i=1,2,\cdots,m\},$$
$$H=\{i \mid a_i = |T\cap B_i| = d_i, i=1,2,\cdots,m\}.$$

T 是划分限制支撑树，当且仅当 $C=D=\varnothing$.

设 T 是 N 的一棵支撑树，则 T 是 N 的最小划分限制支撑树，当且仅当 T 的划分满足：

1° $C=D=\varnothing$;

2° 存在非负数 α_i、β_i，$\alpha_i\beta_i=0$，且满足，

$$\text{若 } \alpha_i>0, \Rightarrow i\in A\cup H,$$
$$\text{若 } \beta_i>0, \Rightarrow i\in B\cup H;$$

3° 在权 $\tilde{w}$ 下，T 是 N 的最小权支撑树；其中对每一边 $e\in B_i, i=1,2,\cdots,m$,

$$\tilde{w}(e) = w(e) - \alpha_i + \beta_i.$$

设 T 是 N 的支撑树，并且关于 T 的划分满足条件1°、2°和3°，证明 T 是 N 的最

小划分限制支撑树.因为

$$\begin{aligned}\tilde{w}(T) &= \sum_{i=1}^{m}\sum_{e\in T\cap B_i}(w(e)-\alpha_i+\beta_i)\\ &= w(T)-\sum_{i=1}^{m}\alpha_i\mid T\cap B_i\mid+\sum_{i=1}^{m}\beta_i\mid T\cap \beta_i\mid\\ &= w(T)-\sum_{i\in A}\alpha_i a_i+\sum_{i\in B}\beta_i d_i+\sum_{i\in H}(\beta_i-\alpha_i),\end{aligned}$$

对 N 的任一划分限制树 T',则

$$\begin{aligned}\tilde{w}(T') &= w(T')-\sum_{i=1}^{m}\alpha_i\mid T'\cap B_i\mid+\sum_{i=1}^{m}\beta_i\mid T'\cap B_i\mid\\ &= w(T')-\sum_{i\in A}\alpha_i\mid T'\cap B_i\mid+\sum_{i\in B}\beta_i\mid T'\cap B_i\mid+\sum_{i\in H}(\beta_i-\alpha_i)a_i.\end{aligned}$$

显然,对 $i\in A$, $\mid T'\cap B_i\mid\geqslant a_i$;对 $i\in B$, $\mid T'\cap B_i\mid\leqslant d_i$;而对 $i\in H$, $\mid T'\cap B_i\mid = a_i = d_i$. 从而

$$\tilde{w}(T')\leqslant w(T')-\sum_{i\in A}a_i\alpha_i+\sum_{i\in B}d_i\beta_i+\sum_{i\in H}(\beta_i-\alpha_i)a_i.$$

由 T 的假设知,

$$0\geqslant\tilde{w}(T)-\tilde{w}(T')\geqslant w(T)-w(T'),\text{即 } w(T)\leqslant w(T').$$

由 T' 的任意性知, T 是 N 的最小划分限制支撑树.

反之,由于 N 的最小划分限制树是下述线性规划的最优解:

$$\begin{cases}\min\sum_{e\in E}w(e)x_e,\\ \sum_{e\in E}x_e = r(E),\\ \sum_{e\in[S]}x_e\leqslant r(S), & S\subset E,\\ a_i\leqslant\sum_{e\in B_i}x_e\leqslant d_i, & i=1,2,\cdots,m,\\ x_e\geqslant 0, & e\in E.\end{cases}\tag{4-4}$$

其中 $[S]$ 表示集合 S 的闭包.见(4-1)式.

(4-4)式的对偶规划为

$$\begin{cases}\max\{r(E)\lambda-\sum_{S\subseteq E}r(S)y_S+\sum_{i=1}^{m}a_i\alpha_i-\sum_{i=1}^{m}d_i\beta_i\},\\ \lambda-\sum_{[S]\ni e}y_S+\alpha_i-\beta_i\leqslant w(e), & e\in B_i, i=1,2,\cdots,m,\\ y_S\geqslant 0, & S\subset E,\\ \alpha_i、\beta_i\geqslant 0, & i=1,2,\cdots,m.\end{cases}\tag{4-5}$$

若记 $\tilde{w}(e)=w(e)-\alpha_i+\beta_i, e\in B_i, i=1,2,\cdots,m$,则(4-5)式可重写为

$$\begin{cases}\max\{r(E)\lambda-\sum_{S\subset E}r(S)y_S+\sum_{i=1}^{m}a_i\alpha_i-\sum_{i=1}^{m}d_i\beta_i,\\ \lambda-\sum_{[S]\in e}y_S\leqslant W(e)-\alpha_i+\beta_i=\tilde{w}(e),\\ y_S\geqslant 0,\qquad S\subset E.\end{cases}\tag{4-6}$$

设 T 是 N 的最小划分限制支撑树，令

$$x_e^0=\begin{cases}1, & 若\ e\in T,\\ 0, & 其余,\end{cases}\tag{4-7}$$

则 $\{x_e^0\}$ 是(4-4) 式的最优解，因此按线性规划原理，(4-5) 式有最优解 $\{\lambda^0,y_S^0,\alpha_i^0,\beta_i^0\}$，使得

$$x_e^0>0\ \Rightarrow\ \lambda^0+\sum_{[S]\ni e}y_S^0=w(e)-\alpha_i^0+\beta_i^0=\tilde{w}(e),\tag{4-8}$$

$$\alpha_i^0>0\ \Rightarrow\ \sum_{e\in B_i}x_e^0=a_i,\qquad 即\ |T\cap B_i|=a_i,$$

$$\beta_i^0>0\ \Rightarrow\ \sum_{e\in B_i}x_e^0=d_i,\qquad 即\ |T\cap B_i|=d_i,$$

$$y_S^0>0\ \Rightarrow\ \sum_{e\in[S]}x_e^0=r(S).\tag{4-9}$$

显然 $\{x_e^0\}$ 是(4-1) 式的可行解，而 $\{\lambda^0,y_S^0\}$ 是(4-6) 式的可行解，并且当在(4-3) 式中取 $w(e)$ 为 $w(e)-\alpha_i^0+\beta_i^0$ 时，$\{\lambda^0,y_S^0\}$ 也是(4-3) 式的可行解. 进而 $\{x_e^0\}$ 和 $\{\lambda^0,y_S^0\}$ 满足互补松弛条件(4-8) 式和(4-9) 式，从而 $\{x_e^0,\lambda^0\}$ 是下述问题的最优解：

$$\begin{cases}\min\sum_{e\in E}\tilde{w}(e)x_e,\\ \sum_{e\in E}x_e=r(E),\\ \sum_{e\in[S]}x_e\leqslant r(S),\quad S\subset V,\\ x_e\geqslant 0.\end{cases}$$

这就意味着 T 满足最小划分限制支撑树的条件 1°、2° 和 3°.

求 N 上最小划分限制树的算法，参见文献[6].

值得注意的是，线性规划(4-4) 有 0-1 整数顶点，从而有 0-1 整数最优解.

另外，最优树形图问题也可化为规划(4-4) 的一种特殊情况，这只要取 $B_j=\{(i,j)\mid i\in V,(i,j)\in A\},j\in V-\{v_0\}$，而 $a_j=d_j=1$ 即可.

上述所有内容几乎都可以扩展到一种抽象的代数结构 —— 拟阵(matroid) 上.

4.7　网络上的施泰纳树

给定欧氏平面上 n 个点 $p_1,p_2,\cdots,p_n$，用线把它们连成一片，要求连线的总长

度最短.这是施泰纳(Steiner)问题的原始含义.最早是在17世纪,由费马(Fermat)提出的 $n=3$ 情形,即任给欧氏平面上三个点 p_1、p_2 和 p_3,求一点 s,使 s 到 p_1、p_2 和 p_3 的距离之和最小.托里析利(Torricelli)证明了下述事实:

(1) 在 $\triangle p_1p_2p_3$ 中,若有一个角大于或等于 $120°$,如 $\angle p_1$,则 $s=p_1$;

(2) 若 $\triangle p_1p_2p_3$ 中每一个角都小于 $120°$,则 s 在 $\triangle p_1p_2p_3$ 的内部,且 $\angle p_1sp_2=\angle p_1sp_3=\angle p_2sp_3=120°$.

到了19世纪初,施泰纳又重新研究这一问题,并把它扩广到一般情况,后来柯朗(Courant)和罗宾斯(Robbins)在 What is Mathematics 一书中,将此问题命名为施泰纳问题.

当 n 比较大时,这一问题十分困难.1990年堵丁柱和黄光明用凸分析的方法,证明了用这 n 个点的最小支撑树的长度 $L_m(n)$,去逼近这 n 个点上的施泰纳问题的最优值 $L_s(n)$,其近似度为 $L_s(n)\geqslant\frac{\sqrt{3}}{2}L_m(n)$.

平面上的施泰纳问题可以扩展到高维空间,也可扩充到曲面上.本节只讨论网络中的施泰纳问题.

设 $N=(V,E;w)$ 是一个连通的无向网络,$R\subset V$ 是 N 的顶点的一个子集.求 N 的一个最小权子树 $T=(U,E')$,使 $R\subseteq U$,$E'\subseteq E$.R 中的顶点称为固定点,$U-R$ 中的顶点称为施泰纳点.在欧氏平面情况下,$|U-R|\leqslant|R|-2$;在一般无向网络情况下,施泰纳点的个数就不好估计,但当 N 是完全图 K_n,并且任意三个点 i、j、k,满足 $w_{ij}+w_{jk}\geqslant w_{ik}$ 时,施泰纳点的个数最多为 $|R|-2$.事实上,此时每一个施泰纳点的次至少为3.由于 n 个点的树恰有 $n-1$ 条边,所以若施泰纳点的个数为 x,则有

$$3x+|R|\leqslant 2(|R|+x-1),$$

即得 $x\leqslant|R|-2$.

由此可建立如下的遍数算法:

步1　求出 N 中所有点对 i、j 之间的最短路及其长度 d_{ij},并以 d_{ij} 为边长构造完全图 K_n.

步2　对每一个子集 $S\subseteq V-R$ 且 $|S|\leqslant|R|-2$,求 K_n 的由 $R\cup S$ 导出子

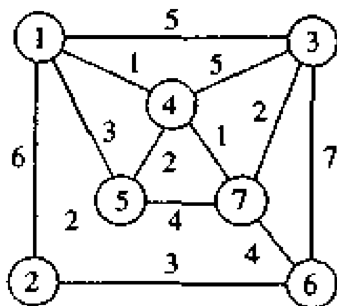


图4-6

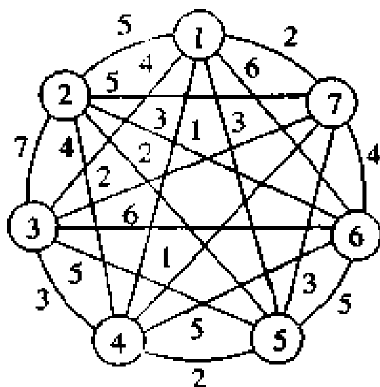


图4-7

图的最小权支撑树 T_S.

步 3　对一切 $S \leqslant V - R$, $|S| \subseteq |R| - 2$, 找出最小权的 T_S, 设它为 T_{*S}. 然后将 T_S^* 还原为 N 的树.

例如, 图 4-6 上 $R = \{1,2,3\}$.

步 1　求所有点对间的最短路, 形成 K_7 如图 4-7 所示.

步 2　由 $|S| \leqslant |R| - 2 = 3 - 2 = 1$, 因此 S 有 5 种可能性, 对每一可能性, 求 $K[R \cup S]$ 上的最小权树如表 4-1.

表 4-1

S	$K[R \cup S]$ 的最小树 T	T 的长度
∅	[1,2],[1,3]	9
{4}	[1,4],[2,4],[3,4]	8
{5}	[1,5],[2,5],[1,3]	9
{6}	[2,6],[1,3],[1,2]	12
{7}	[1,7],[3,7],[1,2]	9

最小树为 $T^* = \{[1,4],[2,4],[3,4]\}$.

步 3　把 T^* 中每一边换为 N 中的一条路: 如 T^* 中的边[1,4]在 N 中的路为([1,4]);

T^* 中的边[2,4]在 N 中的路为([2,5],[5,4]);

T^* 中的边[3,4]在 N 中的路为([3,7],[7,4]).

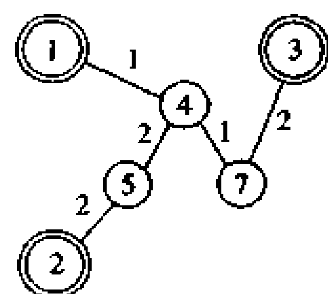


图 4-8

从而 N 中支撑 R 的施泰纳最小树为图 4-8.

这个算法的计算复杂性为 $O(|R|^2 2^{|S|} + |V|^3)$, 因此它是 $|S|$ 的指数函数.

有向网络 $N = (V, A; w)$ 上的施泰纳树问题, 可叙述如下: 设 $r_0 \in V$ 为一指定点, 称之为根. 给定一个子集 $R \subseteq V - \{r_0\}$, 求 N 的一棵以 r_0 为根的子树形图 T, 使 T 包含 R 中所有点, 并且 T 的权最小. 显然, 当 $|R| = 1$ 时, 等价于求 N 中自 r_0 到 R 的一条最短路; 而当 $R = V - \{r_0\}$ 时, 则等价于求 N 的以 r_0 为根的最小树形图. 对一般的 R, 这个问题同样是十分困难的. 已知它可以描述为下述三种混合整数规划模型:

$$
(\mathrm{P}_1)\quad \begin{cases} \min \sum\limits_{(i,j)\in A} w_{ij} y_{ij}, \\ \sum\limits_{j\in\alpha(i)} x_{ij} - \sum\limits_{j\in\beta(i)} x_{ji} = \begin{cases} -1, & i \in R, \\ 0, & i \in V - \{\{r_0\} \cup R\}, \end{cases} \\ 0 \leqslant x_{ij} \leqslant |R|\, y_{ij}, \quad (i,j) \in A, \\ y_{ij} \in \{0,1\}, \quad (i,j) \in A. \end{cases}
$$

这一模型是网络流与选址问题的结合, 对每一点 $i \in R$, 其收量为 1, 而 r_0 为发

点,其余各点是中间点,即流入量等于流出量. $y_{ij}=1$ 表示弧(i,j)被选取.

$$(\mathrm{P}_2)\quad\begin{cases}\min \sum\limits_{(i,j)\in A} w_{ij}y_{ij},\\ \sum\limits_{j\in\alpha(i)} x_{ij}^{(k)}-\sum\limits_{j\in\beta(i)} x_{ji}^{(k)}=\begin{cases}-1, & i=k\in R,\\ 0, & i\in V-\{r_0,k\},k\in R,\end{cases}\\ 0\leqslant x_{ij}^{(k)}\leqslant y_{ij}, \quad (i,j)\in A,\\ y_{ij}\in\{0,1\}, \qquad (i,j)\in A.\end{cases}$$

这是多种物资流和场址选择问题的结合,同样,r_0 为发点,它有 $|R|$ 种物资,每一种物资只有一个单位;R 为收点集,收点 $k\in R$ 只能收一个单位的第 k 种物资. y_{ij} 与模型(P_1)相同.

$$(\mathrm{P}_3)\quad\begin{cases}\min \sum\limits_{(i,j)\in A} w_{ij}y_{ij},\\ \sum\limits_{(i,j)\in(X,\overline{X})} y_{ij}\geqslant 1, \quad \text{对一切 } R \text{ 割集}(X,\overline{X}),\\ y_{ij}\in\{0,1\}, \qquad (i,j)\in A.\end{cases}$$

这儿 R 割集$(X,\overline{X})$的定义为:$X\subset V,\overline{X}=V-X,(X,\overline{X})=\{(i,j)\in A\mid i\in X,j\in\overline{X}\}$,并且 $r_0\in X,R\cap\overline{X}\neq\varnothing$.

关于施泰纳问题的更深入的结果,参看 F.K.Hwang 和 D.S.Richards 的综述文章.

5 网络中的选址问题

选址问题也称为集散中心问题,它是生产实际中一类非常广泛且十分重要的问题.例如工厂位置选择,城市里的消防站的位置选择,医院位置的确定,急救中心位置的确定等等,都属于选址问题.评价一个场址的优劣有许多标准,为了使问题简单而又有意义,通常以经济效益为标准,而经济上的效益又往往把交通运输上的费用视为重点.常用的有两类评价(或目标)函数:权和最小;最大的最小.所谓权和最小,是指场址到用户的加权距离和最小;最大的最小是指场址到用户的最大距离最小.

选址问题可以分为两大类:连续型的和离散型.前者多半是用解析方法,后者主要用组合性方法,而网络中的选址问题属于离散型.

从另一角度看,场址问题可分为单场址和多场址两类.多场址问题很困难,至今没有多少好的结果.故本章只讨论网络中的单场址问题.

5.1 网络的重心

设 $N=(V,E;w,l)$ 为一赋权网络.其中 $w:V\to\mathbf{R}^+,l:E\to\mathbf{R}^+$,即对每一顶点

$i \in V$,有一个权 w_i,而对每一边$[i,j]$,有一长度 l_{ij}.网络重心问题是求一点 $x \in V \cup E$,使

$$f(x) = \sum_{i \in V} w_i d(i,x) \tag{5-1}$$

最小.这儿 $x \in V \cup E$ 表示 x 可以是顶点,也可以在边上;$d(i,x)$ 表示顶点 i 到重心 x 的距离.

不难证明,使(5-1)式达到最小的 x 必能在 V 中找到,即

$$\min_{x \in V \cup E} f(x) = \min_{x \in V} f(x). \tag{5-2}$$

事实上,假如最优位置 x^* 在边$[p,q]$上.设 x^* 到 p 点的距离为 d,到 q 点的距离为 $l_{pq} - d$;那么 V 中点可分为两部分 V_p 和 V_q,V_p 中点到 x^* 的最短路经过 p 点,而 V_q 中点到 x^* 的最短路经过 q 点.令

$$w(V_p) = \sum_{i \in V_p} w_i, \quad w(V_q) = \sum_{i \in V_q} w_i,$$

不妨设 $w(V_p) \leqslant w(V_q)$,则

$$\begin{aligned} f(x^*) &= \sum_{i \in V} w_i d(i,x^*) = \sum_{i \in V_p} w_i d(i,p) + w(V_p)d + \\ &\quad \sum_{i \in V_q} w_i d(i,q) + w(V_q)(l_{pq} - d) \\ &\geqslant \sum_{i \in V_p} w_i d(i,p) + w(V_p)d + \sum_{i \in V_q} w_i d(i,q) + w(V_p)(l_{pq} - d) \\ &= \sum_{i \in V_p} w_i (d(i,p) + l_{pq}) + \sum_{i \in V_q} w_i d(i,q) \\ &\geqslant \sum_{i \in V} w_i d(i,q) \geqslant \min_{x \in V} \sum_{i \in V} w_i d(i,x) = \min_{x \in V} f(x), \end{aligned}$$

这就证明了(5-2)式.

由(5-2)式可建立求网络重心的算法如下:

步1　求网络 $N = (V,E;w,l)$ 中所有点对间的最短路,因此得最短路长度矩阵 $\boldsymbol{D} = [d(i,j)]$.

步2　求

$$f(x^*) = \min_{x \in V} \sum_{i \in V} w_i d(i,x),$$

x^* 即为网络 N 的重心.

当网络 N 是一棵树时,此时 N 中任意两点间有唯一的一条路.由(5-2)式的证明可见,此时重心的位置只和顶点上的权有关,所以当 N 为树时,求树的重心有非常简单的算法:

步0　计算树 N 上各点权和之半,即

$$s = \frac{1}{2} \sum_{i \in V} w_i.$$

步1　在 N 中任取一个端点 i,若 $w_i \geqslant s$,则顶点 i 为 N 的重心,否则进行第2步.

步 2 设顶点 i 的唯一邻点为 j，置 $N \leftarrow N - \{i\}, w_j \leftarrow w_j + w_i$，返回第 1 步.

对于有向网络 $N = (V, A; w, l)$ 而言，由于路是有向的，l_{ij} 和 l_{ji} 可能不等，所以可能 $d(i,j) \neq d(j,i)$. 此时，满足下式的 x^* 称为 N 的**发射重心**.

$$f^+(x^*) = \min_{x \in V} \sum_{i \in V} w_i d(x, i).$$

而满足下式的 x^0、称为 N 的**汇集重心**：

$$f^-(x^0) = \min_{x \in V} \sum_{i \in V} w_i d(i, x).$$

显然，发射重心和汇集重心一般地是不会重合的. 其求解方法类似于无向网络求重心的方法.

值得注意的是，对有向网络仍有

$$\min_{x \in V \cup A} \sum_{i \in V} w_i d(x, i) = \min_{x \in V} \sum_{i \in V} w_i d(x, i),$$

$$\min_{x \in V \cup A} \sum_{i \in V} w_i d(i, x) = \min_{x \in V} \sum_{i \in V} w_i d(i, x).$$

5.2 网络的中心

设 $N = (V, E; w, l)$ 为无向网络，所谓 N 的**网络中心**，是指下述问题的最优解 x^*：

$$\min_{x \in V} \max_{i \in V} w_i d(i, x), \tag{5-3}$$

其中 $d(i, x)$ 表示 N 中点 i 到顶点 x 的最短路长度.

由(5-3)式不难得到求 N 的中心的算法：

步 0 利用最短路算法求出所有点对间的最短路长度矩阵 $\boldsymbol{D} = [d(i,j)]$.

步 1 计算

$$g(x) = \max_{i \in V} \{w_i d(i, x)\}, \quad x \in V.$$

步 2 求 $\min\{g(x) \mid x \in V\} = g(x^*)$

则 x^* 为 N 的中心.

无向网络 $N = (V, E; w, l)$ 的**绝对中心**，是指下述问题的最优解 x^*：

$$\min_{x \in V \cup E} \max_{i \in V} w_i d(i, x). \tag{5-4}$$

设 $[i, j]$ 为 N 中一条边，x_{ij} 为边 $[i, j]$ 上一个动点，它也表示该点离点 i 的距离. 定义 N 中点 $k \in V$ 到 x_{ij} 的距离为

$$r_k(x_{ij}) = \min\{d(k, i) + x_{ij}, d(k, j) + l_{ij} - x_{ij}\},$$

其中 $d(k, i)(d(k, j))$ 表示顶点 k 到顶点 $i(j)$ 的最短路长度.

对每一顶点 $k \in V$ 和每一边 $[i, j]$，计算

$$\begin{aligned} \min_{0 \leqslant x_{ij} \leqslant l_{ij}} r_k(x_{ij}) &= \min_{0 \leqslant x_{ij} \leqslant l_{ij}} \min\{d(k, i) + x_{ij}, d(k, j) + l_{ij} - x_{ij}\} \\ &= r_k(x_{ij}^k), \qquad k \in V, [i, j] \in E, \end{aligned}$$

求

$$R(x_{ij}^{0}) = \max_{k\in V}\{r_k(x_{ij}^{k})w_k\},\quad [i,j]\in E,$$

和最后求

$$\min_{[ij]\in E}\{R(x_{ij}^{0})\} = R(x_{pq}^{0}),$$

则绝对中心为边$[p,q]$上的点 x_{pq}^{0}.

总结上述,可得如下算法:

步0　利用最短路算法,求出 N 中所有点对之间的最短路长度矩阵$\boldsymbol{D}=[d(i,j)]$.

步1　对每一个点 k 和每一条边$[i,j]$,计算

$$\min_{0\le x_{ij}\le l_{ij}} r_k(x_{ij}) = \min_{0\le x_{ij}\le l_{ij}}\min\{d_{ki}+x_{ij}, d_{kj}+l_{ij}-x_{ij}\} = r_k(x_{ij}^{k}).$$

步2　对每一边$[i,j]\in E$,求

$$R(x_{ij}^{0}) = \max_{k\in V}\{r_k(x_{ij}^{k})\}.$$

步3　计算$\min\limits_{[i,j]\in E}\{R(x_{ij}^{0})\} = R(x_{pq}^{0})$.

5.3　网络的形心

在5.1节中的重心问题,可视为服务对象是网络中的顶点,而最优的服务重心一定可设在顶点上;而5.2节中的中心问题,也是把顶点视为服务对象,而服务中心要求设在顶点上;网络的绝对中心,是把顶点视为服务对象,而服务中心可设在边上.

本节讨论**网络形心问题**,是把边视为服务对象(如管道和线路的维修等),而服务站要求设在网络的顶点上.

设 $N=(V,E;l)$ 为无向网络,对每一边 $e\in E$ 有一权$l(e)\geq 0$. N 中一个点k到一条边 e 的距离,定义为 $r_k(e)$,其中 $e=[i,j]$,

$$r_k(e) = \frac{1}{2}(d_{ki}+d_{kj}+l(e)),$$

这儿 $d_{ki}(d_{kj})$ 表示 N 中点k到点$i(j)$ 的最短路长度.

N 的**网络形心**为满足下式的极小顶点 k^*:

$$\min_{k\in V}\ \max_{e\in E} r_k(e).$$

N 的**绝对形心**定义为使

$$\min_{x\in V\cup E}\max_{e\in E} r_x(e)$$

达到最小的点 x^*. 由于绝对形心计算起来过于复杂,在此省略. 下面仅给出形心的算法:

步0　用最短路算法求出 N 中所有点对之间的最短路长度,得 $n\times n$ 距离矩阵 $\boldsymbol{D}=[d_{ij}]$.

步1　对每一顶点 $k\in V$ 和每一边 $e_t\in E$,计算 $r_k(e_t)$. 这样得到一个 $n\times m$ 的矩阵 $\boldsymbol{R}=[r_k(e_t)]$,其中 $n=|V|, m=|E|$. 易见

$$R = \frac{1}{2}(DB(N) + J_nL),$$

其中 $B(N)$ 为 N 的关联矩阵；J_n 是分量全为 1 的 n 维列向量. $L = (l(e_1), l(e_2), \cdots, l(e_m))$ 为边权向量.

步 2 在 R 的每一行中取一个最大元素，然后在各行的最大元素中取一个最小者. 那么这个最小元素所对应的顶点即为 N 的形心.

5.4 离散的多场址问题

在生产实际中，一般地说，可供选择的场址是有限个，而不是每个地方都可建厂的. 因此问题变为在有限几个场址中，选其中一个或几个. 一般模型如下：

设某种产品有 n 个需求地，第 i 个需求地的需求量为 a_i；该产品有 m 个工厂进行生产，第 j 个工厂的产量为 b_j；由于供不应求，即 $\sum_{i=1}^{n} a_i > \sum_{j=1}^{m} b_j$，现需要再建 k 个工厂生产该产品，已知有 r 个场地可供选择（即 $r > k$）. 假如限制新建的 k 个工厂的生产能力均相同，即均为 $a = \frac{1}{k}\left(\sum_{j=1}^{m} b_j - \sum_{i=1}^{n} a_i\right)$. 在已知工厂 i 到需求地 j 的单位产品的运费为 c_{ij}，以及场地 l 到需求地 j 的单位产品运费为 d_{lj} 的情况下，确定新建 k 个厂的位置，使总的运费最小.

设第 i 工厂运往第 j 个需求地的产品数量为 x_{ij}；第 l 个场地（若建厂）运往第 j 个需求地的产品数量为 y_{lj}；设 $z_l = 0,1$，$z_l = 1$ 表示在场地 l 建厂，$z_l = 0$ 表示在场地 l 不建厂. 那么上述问题可化为下述模型：

$$\begin{cases} \min \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{l=1}^{r}\sum_{j=1}^{m} d_{lj}y_{lj}, \\ \sum_{j=1}^{m} x_{ij} = a_i, \quad i = 1,2,\cdots,n, \\ \sum_{j=1}^{m} y_{lj} = az_l, \quad l = 1,2,\cdots,r, \\ \sum_{i=1}^{n} x_{ij} + \sum_{l=1}^{r} y_{lj} = b_j, \quad j = 1,2,\cdots,m, \\ \sum_{l=1}^{r} z_l = k, \\ x_{ij}, y_{lj} \geqslant 0, \quad i = 1,2,\cdots,n, j = 1,2,\cdots,m, \\ z_l = 0,1, \quad l = 1,2,\cdots,r. \end{cases}$$

这是一个混合整数规划问题. 一般地很难求解. 但当 $r - k$ 不是太大时，可以通过求解 C_r^k 个运输问题即可求出这 k 个工厂的位置.

6 网络中的最优匹配

6.1 基本概念

设 $N=(V,E;w)$ 是一个无向网络，对每一边 e 有一个非负权 $w(e)$. 设 M 是 N 中边的一个子集，若 M 中任何两条边无公共端点，则称 M 为 N 的一个**匹配**. N 中具有边数最多的匹配，称为 N 的**最大匹配**. 一个匹配 M 的权，定义为 M 中边的权之和. N 中具有权最大的匹配，称为 N 的**最大权匹配**. 设 M 是 N 的一个匹配，v 是 N 的一个顶点，若 v 不与 M 中任何边相关联，则称 v 为 M 的**未盖点**，否则称为 M 的**已盖点**；若 N 中不存在 M 的未盖点，则称 M 为 N 的**完美匹配**. 显然完美匹配一定是最大匹配.

设 M 是 N 的一个匹配，P 是 N 的一条链，若 P 中的边是交替地属于 M 和 $E-M$，则称 P 为**交错链**；若交错链 P 的两个端点均是 M 的未盖点，则称 P 为 M 的**增广交错链**. 显然，M 是 N 的最大匹配，当且仅当 N 中不存在关于 M 的增广交错链. 根据这一结论可建立求最大匹配的算法. 然而，如何找增广交错链？这是匹配算法的关键.

最大匹配和最大权匹配，在二部图的情况可用最大流和最小费用流的方法求得. 对非二部图情况，1965 年 J. Edmonds 给出了一个强多项式算法. 算法的基础是线性规划的原始对偶间的互补最优性原理.

6.2 最大基数匹配算法

设 $G=(V,E)$ 是一无向图，记 $V=\{1,2,\cdots,n\}$.

步 0 置 $k\leftarrow 0, G_k\leftarrow G, M_k\leftarrow\varnothing$.

步 1 对 G_k 中每一个 M_k 的未盖点给一个标号 $(S,\varnothing)$（其中第一个标号 S 表示外点，T 表示内点；第二个标号表示由哪一个点而得到的标号），并且所有标号点均未检查.

(1-1) 若所有标号点都检查完了，转步 4. 否则找一个标号未检查的点 i，若 i 有 S 标号（即外点），进行(1-2)；若 i 有 T 标号（即内点），进行(1-3).

(1-2) 检查 S 标号点 i 如下：对每一个边 $[i,j]\in M_k$，若点 j 有 S 标号，则自 i 和 j 分别按其第 2 个标号返回追踪. 若它们的标号来自于两个不同的根（M_k 的未盖点），则转步 2，若它们的标号来自同一个根，则转步 3；若 j 未标号，则给 j 标以 (T,i). 对点 i 的检查完成后，返回(1-1)步.

(1-3) 检查 T 标号点 i 如下：对唯一边 $[i,j]\in M_k$，若 j 有 T 标号，则分别由 i 和 j 的第二标号返回追踪. 若他们的标号来自不同的根，则进行步 2；若来自相同的根，则进行步 3；若 j 未标号，则给 j 标以 (S,i). 返回(1-1).

步 2　找出关于 M_k 的增广交错链 P，并置 $M_k \leftarrow M_k \oplus P$，其中 $M_k \oplus P = M_k \cup (P - M_k) \cap P$. 然后去掉所有标号返回步 1.

步 3　找出奇圈 C_k 及由根到 C_k 的交错链上与 C_k 相交的第一个点，记为 b_{C_k}（称 b_{C_k} 为 C_k 的蒂），在 G_k 中把 C_k 收缩为一点，仍记为 b_{C_k}，这样由 G_k 得到一个图，记为 G_{k+1}. 收缩点 b_{C_k} 称为 G_{k+1} 的**拟点**. 若 G_k 中 b_{C_k} 为外（内）点，则在 G_{k+1} 中 b_{C_k} 称为**外（内）拟点**. 在 G_k 的 C_k 中可能有拟点，这些拟点在 G_{k+1} 中称为**内层拟点**. 在 G_{k+1} 中拟点 b_{C_k} 作为标号未检查点看待，其标号与它在 G_k 中相同. 置 $M_{k+1} \leftarrow M_k - C_k$.

置 $k \leftarrow k+1$，根据 b_{C_k} 是外点还是内点，返回(1-2)或(1-3)，对 b_{C_k} 继续检查.

步 4　所有标号点都检查完了，且不存在关于 M_k 的增广交错链. 此时 M_k 是 G_k 的最大匹配. 把所有拟点按后收缩先展开的原则还原为奇圈，并适当地选取奇圈 C_{k-1} 上的一个最大匹配 N_{k-1}，使 $N_{k-1} \cup M_k$ 为 G_{k-1} 的匹配 M_{k-1}. 若 $k-1>0$，重复这个过程.

例 1　求图6-1(a)的最大匹配. 已知匹配 $M_0 = \{[2,4],[6,8]\}$.

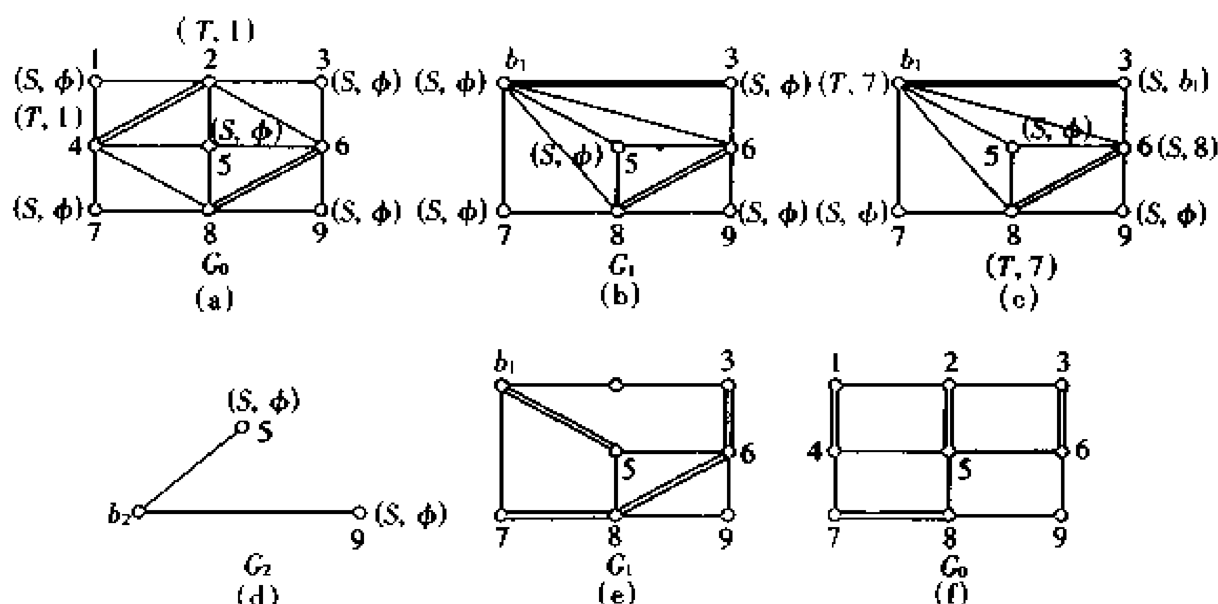


图 6-1

首先，对一切 M_0 的未盖点均标以 $(S,\varnothing)$. 选取标号未检查的点 1 进行检查. 从而 2 和 4 两点得标号 $(T,1)$，这样点 1 检查完. 此时标号未检查的点集为 $\{2,3,4,5,7,9\}$. 取点 2 检查. 由 2 的标号为 T 标号，由 $[2,4] \in M_0$ 且 4 也是 T 标号，故由 2 和 4 点返回追踪，它们的标号都是点 1，从而得奇圈 $C_0 = (1,2,4)$，或记 $C_0 = \{[1,2],[2,4],[4,1]\}$. 将 C_0 缩为一点 b_1. 由于 C_0 的蒂是点 1 且是外点，所以 b_1 也是外点. 故在 G_1 上继续检查 b_1. 此时 $M_1 = \{[6,8]\}$，从而 $[b_1,3]$ 是 G_1 的增广交错链. 这样 M_1 扩大为 $\{[b_1,3],[6,8]\}$. 此时未盖点为 7、5、9. 选取点 7 进行检查，b_1 和 8 得标号 $(T,7)$. 再检查 b_1 和 8 时，3 得标号 (S,b_1)，6 得标号 $(S,8)$；再检查点 3 时，得奇圈 $C_1 = \{[7,b_1],[b_1,3],[3,6],[6,8],[8,7]\}$. 收缩 C_1 得图 G_2，此时 $M_2 = \varnothing$. 再继续检

查 b_2,得增广交错链$[b_2,5]$,它也是 G_2 的最大匹配.然后进行第4步,依次把 b_2 和 b_1 还原为奇圈,并选取圈上适当的最大匹配,见图(e)和(f).

6.3 最大权匹配

设 $N=(V,E;w)$ 是一个赋权无向网络,即 N 中每一边$[i,j]\in E$有一非负权 w_{ij}. N 的一个匹配 M 的权定义为 $w(M)=\sum_{e\in M}w(e)$. 使权最大的匹配,称为**最大权匹配**.

首先把 N 的最大权匹配问题描述为线性规划.

对 N 中每一边 $e=[i,j]\in E$ 定义一个变量 $x(e)$. 设 $S_1,S_2,\cdots,S_t$ 为 N 的所有奇数个点的集合. 用$[S_i]$表示 S_i 在 N 中的导出子图. 用 E_j 表示 N 中与点 j 关联的边之集合. 那么下述不等式组刻画了 N 的匹配:

$$\begin{cases}\sum_{e\in E_i}x(e)\leqslant 1, & i\in V,\\ \sum_{e\in[S_j]}x(e)\leqslant\frac{1}{2}(|S_j|-1), & j=1,2,\cdots,t,\\ x(e)=0\text{ 或 }1, & e\in E.\end{cases}\tag{6-1}$$

容易验证,(6-1)式的任一个解$\{x^0(e)\}$,若定义 $M=\{e\mid x^0(e)=1\}$,则 M 是 N 的一个匹配;反之,若 M 是 N 的一个匹配,则令

$$x^0(e)=\begin{cases}1, & \text{若 } e\in M,\\ 0, & \text{否则},\end{cases}$$

那么$\{x^0(e)\}$是(6-1)式的一个解. 有趣的是(6-1)式的所有解构成的多面体具有0,1坐标的整数顶点. 因此,最大权匹配问题可化为线性规划:

$$(\mathrm{P})\quad\begin{cases}\max\sum_{e\in E}w(e)x(e),\\ \sum_{e\in E_i}x(e)\leqslant 1, & i\in V,\\ \sum_{e\in[S_j]}x(e)\leqslant\frac{1}{2}(|S_j|-1), & j=1,2,\cdots,t,\\ x(e)\geqslant 0, & e\in E.\end{cases}$$

然而这个线性规划的约束数量随 N 的顶点个数增加而指数地增加,故线性规划的单纯形算法实际上是不适用的. (P)的对偶问题为

$$(\mathrm{D})\quad\begin{cases}\min\sum_{i\in V}\left\{y_i+\frac{1}{2}\sum_{j=1}^{t}(|S_j|-1)z_j\right\},\\ y_i+y_j+\sum_{[S_k]\ni[i,j]}z_k\geqslant w(i,j), & [i,j]\in E,\\ y_i,z_k\geqslant 0, \quad i\in V,k=1,2,\cdots,t.\end{cases}$$

线性规划的对偶理论告诉我们：若 $\{x(e)\}$ 和 $\{y_i, z_k\}$ 分别为(P) 和(D) 的可行解，那么它们分别为(P) 和(D) 的最优解，当且仅当下述条件成立：

(1) $x([i,j]) > 0 \Rightarrow y_i + y_j + \sum_{S_k \ni [i,j]} z_k = w(i,j)$.

(2) $y_i > 0 \Rightarrow \sum_{e \in E_i} x(e) = 1$.

(3) $z_k > 0 \Rightarrow \sum_{e \in [S_k]} x(e) = \frac{1}{2}(|S_k| - 1)$.

求最大权匹配的算法，是从(P) 和(D) 的可行解 $\{x(e)\}$ 和 $\{y_i, z_k\}$ 且满足条件(1) 和(3) 出发，逐渐增大匹配；当匹配不能增加时，修正对偶可行解 $\{y_i, z_k\}$，使之在保证条件(1) 和(3) 成立的情况下，条件(2) 逐步满足. 通过交替地增加匹配和改善对偶可行解的办法，使最终得到的可行解 $\{x(e)\}$ 以及 $\{y_i, z_k\}$ 满足所有条件(1)、(2) 和(3). 这样就得到(P) 的最优解 $\{x(e)\}$，它就对应于 N 的最大权匹配.

由于求非二部图最大权算法十分复杂，且本文篇幅有限，下面只给出算法的梗概，详细的计算步骤请参见文献[4].

步 0　设 M 是网络 N 的一个匹配，开始可取 $M = \varnothing$. 取 $y_i = \frac{1}{2}\max\{w(e) \mid e \in E\}$, $z_k = 0$, $i \in V$. 记 $\bar{w}_{ij} = y_i + y_j + \sum z_k - w_{ij}$. $\bar{E} = \{[i,j] \in E \mid \bar{w}_{ij} = 0\}$.

步 1　以每一个未盖点 v 为根，在 $(V, \bar{E})$ 中用标号法找增广交错链（具体方法与最大匹配算法步 1 类同）. 若找到了增广链，转步 2；若发现了奇圈，转步 3；否则转步 4.

步 2　找出增广链（若链上有收缩点，要通过被收缩的点构造出增广链），增大匹配. 除去点和收缩点上所有标号，返回步 1.

步 3　找出奇圈，把它收缩为一点，并给它 S 标号. 它对应的对偶变量 z_k 置为 0，返回步 1.

步 4　改变对偶变量：取

$\delta_1 = \min\{y_i \mid$ 对一切 S 标号点 $i\}$,

$\delta_2 = \min\{y_i + y_j - w_{ij} \mid [i,j] \in E$, i 有 S 标号且 j 是未标号 T 点$\}$,

$\delta_3 = \frac{1}{2}\min\{y_i + y_j - w_{ij} \mid [i,j] \in E$, i 和 j 是 S 标号点或者最外层奇圈上 S 标号点$\}$,

$\delta_4 = \frac{1}{2}\min\{z_k \mid$ 最外层奇圈对应的收缩点 k 有 T 标号$\}$,

$\delta_5 = \min\{y_i + y_j - w_{ij} \mid [i,j] \in E$, i 有 S 标号或包含在有 S 标号收缩点的最外层奇圈上，j 是未标号或者含在未标号收缩点的最外层奇圈上$\}$.

令 $\delta = \min\{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5\}$，置

$$y_i = \begin{cases} y_i + \delta, & \text{若 } i \text{ 为内点}, \\ y_i - \delta, & \text{若 } i \text{ 为外点}. \end{cases}$$

$$z_k = \begin{cases} z_k + 2\delta, & \text{若收缩点 } k \text{ 为外点}, \\ z_k - 2\delta, & \text{若收缩点 } k \text{ 为内点}. \end{cases}$$

返回步 1.

6.4　中国邮递员问题

一个邮递员每天都要从邮局出发，沿他所管辖的街道逐户送报，然后返回邮局. 如何选取其行走路线，使其所走的总路程最短？这一问题是由山东师范大学管梅谷先生在 20 世纪 60 年代初提出来的. 后来被国际上称为**中国邮递员问题**. 用网络的语言可描述如下：给定一个连通(强连通)无向(有向)网络 $N=(V,E;w)$，对每一边(弧)$[i,j]$((i,j))有一权 $w_{ij}\geqslant 0$. 从 N 中一个顶点出发，沿网络中的边(弧)行走，使每条边(弧)至少经过一次，然后回到出发点，这样一条路线称为投递路线. 投递路线的长度定义为依次经过的边(弧)的长度之和. 长度最短的投递路线，称为**最优投递路线**. 问题是如何找出一个网络的最优投递路线.

显然，当给定的网络是连通的欧拉图时，则它的任何一条欧拉圈(回路)都是最优的投递路线(关于欧拉图的定义见图论篇). 因此只有网络为非欧拉图时，中国邮递员问题才有意义. 由此可重新定义中国邮递员问题为：对给定的连通(强连通)无向(有向)网络 $N=(V,E;w)$，求 E 的一个子集 $M\subseteq E$，使 $\tilde{N}=(V,E+M;w)$ 为欧拉图，并使 $w(M)=\sum\limits_{[ij]\in M}w_{ij}$ 最小. 使 $\tilde{N}$ 为欧拉图的 M，称为 N 的可行加边(弧)集. 权和最小的可行加边(弧)集，称为最优的.

6.4.1　无向网络的邮递员问题

设 $N=(V,E;w)$ 为无向连通网络，M 是 N 的一个可行加边集，则 M 是最优的，当且仅当对 N 中每一个圈 C，有

$$w(C\cap M)\leqslant \frac{1}{2}w(C),$$

其中 $w(Y)=\sum\limits_{e\in Y}w(e)$.

由此结论可得如下算法：

步 0　设 N 是连通的非欧拉网络. 否则或者无解或者是平凡的. 设 M 是 N 的任一个可行加边集.

步 1　若对 N 中每一个圈 C，均有

$$w(C\cap M)\leqslant \frac{1}{2}w(C),$$

算法终止. $\tilde{N}=(V,E+M)$ 中每一条欧拉圈都是最优的投递路线.

步 2　若有一圈 C，使得

$$w(C\cap M)> \frac{1}{2}w(C),$$

则置 $M \leftarrow (C - M) \cup (M - C)$,返回步 1.

由于 M 的权每次迭代都是严格减小的,并且始终保证是可行加边集,故算法的有限性是显然的.然而要迭代多少次?它与网络中圈的数目有关,而圈的数目有可能是顶点数目的指数函数.这样一来,算法就不是多项式时间的了.1965 年 J. Edmonds 把这一问题化为匹配问题去解,从而得到一个强多项式时间的算法:

步 0　设 $N = (V, E; w)$ 是一个连通无向非欧拉网络.令 X 为 N 中所有奇次顶点的集合.在 N 中求出 X 中所有点对之间的最短路 $\{P_{ij} \mid i, j \in X\}$ 及其长度 $w(P_{ij})$.

步 1　构造完全图 K_r. K_r 的顶点集为 X, $i, j \in X$; K_r 中边 $[i, j]$ 的长度为 $w(P_{ij})$,其中 $r = |X|$ 为偶数(因为一个图的奇次顶点个数必为偶数).

步 2　利用最小权匹配算法求出 K_r 的最小权完美匹配 $\bar{M}$.

步 3　将 $\bar{M}$ 中每一条边 $[i, j]$ 还原为 N 中 i 和 j 之间的最短路 P_{ij},那么 $M = \bigcup_{[i,j] \in \bar{M}} P_{ij}$ 即为 N 上中国邮递员问题的最优可行加边集.

6.4.2　有向网络最优投递路线

有向网络中的中国邮递员问题,可利用最小费用流的方法求出最优投递路线或最优加弧集.

步 0　设 $N = (V, A; w)$ 是强连通的有向网络.若对每一点 $v \in V$ 有 $d^+(v) = d^-(v)$,则 N 为欧拉的.此时 N 的每一条欧拉回路都是最优投递路线.设 N 非欧拉.

步 1　构造新的网络 $\tilde{N} = (\tilde{V}, \tilde{A}; \tilde{c}, \tilde{w})$ 如下:

$$\tilde{V} = \{s, t\} \cup S \cup T \cup R, \quad \tilde{A} = A \cup A_1 \cup A_2,$$

其中 $S = \{v \in V \mid d^-(v) - d^+(v) > 0\}$, $T = \{v \in V \mid d^+(v) - d^-(v) > 0\}$,

$$R = \{v \in V \mid d^-(v) - d^+(v) = 0\},$$

$$A_1 = \{(s, i) \mid i \in S\}, \quad A_2 = \{(j, t) \mid j \in T\},$$

$$\tilde{w}_{ij} = \begin{cases} w_{ij}, & 若(i,j) \in A, \\ 0, & 若(i,j) \in A_1 \cup A_2, \end{cases} \qquad \tilde{c}_{ij} = \begin{cases} d^-(j) - d^+(j), & (i,j) \in A_1, \\ d^+(i) - d^-(i), & (i,j) \in A_2, \\ \infty, & (i,j) \in A. \end{cases}$$

这儿 $\tilde{c}_{ij}$ 和 $\tilde{w}_{ij}$ 表示 $\tilde{N}$ 中弧 (i, j) 的容量和通过单位流的费用. s 和 t 分别为 $\tilde{N}$ 的发点和收点.

步 2　在 $\tilde{N}$ 中求自 s 到 t 的最小费用的最大流,设这个流为 $\{f^0(i, j)\}$.

步 3　在 N 中,将每一弧 $(i, j) \in A$ 变为 $f^0(i, j) + 1$ 条平行弧.这样由 N 得到一个欧拉网络 N^0. N^0 中每一条欧拉回路都是最优的投递路线.

7 网络中的逆最优化问题

网络中的逆最优化问题，起始于两个比利时人 Burton 和 Toint，他们首先提出了最短路问题的逆问题. 其提法如下：给定有向网络 N 中 s 到 t 的一条路，通过调整网络中弧的权，使其变为 s 到 t 的最短路. 问题是如何调整其权，使给定的 s 到 t 的路变为最短路时总的调整量最小. 其物理意义是通过调整各路段的票价，使旅客分流，从而减少某些路线的交通拥挤现象.

1994 年张建中等人开始了这方面的研究，并得到了若干极有价值的结果.

首先给出一个最优化问题的逆问题的定义.

设有最优化问题：

$$\min\{f(\boldsymbol{c},\boldsymbol{x}) \mid \boldsymbol{x} \in D\}, \tag{7-1}$$

其中 $\boldsymbol{c} \in \mathbf{R}^n$ 为一参数向量，D 为可行域，$\boldsymbol{x}$ 是未知向量；问题(7-1)的最优解与参数向量 $\boldsymbol{c}$ 有关，并且问题(7-1)的任何一个可行解都可以通过改变 $\boldsymbol{c}$ 而变为最优解.

设 $\boldsymbol{x}^0 \in D$，定义

$$F(\boldsymbol{x}^0) = \{\bar{\boldsymbol{c}} \in \mathbf{R}^n \mid \min\{f(\bar{\boldsymbol{c}},\boldsymbol{x}) \mid \boldsymbol{x} \in D\} = f(\bar{\boldsymbol{c}},\boldsymbol{x}^0)\},$$

那么关于 $\boldsymbol{x}^0$ 的问题(7-1)的逆问题定义为

$$\min\{\|\boldsymbol{c} - \bar{\boldsymbol{c}}\| \mid \bar{\boldsymbol{c}} \in F(\boldsymbol{x}^0)\}, \tag{7-2}$$

这儿 $\|\boldsymbol{c} - \bar{\boldsymbol{c}}\|$ 表示 $\boldsymbol{c} - \bar{\boldsymbol{c}}$ 的范数，通常选取范数 l_1、l_2 或 l_∞.

7.1 线性规划的逆问题

给定标准线性规划

$$(\mathrm{LP})\quad \begin{cases} \min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \\ \quad A\boldsymbol{x} = \boldsymbol{b}, \\ \quad \boldsymbol{x} \geqslant 0, \end{cases}$$

其中 A 是 $m \times n$ 实矩阵，$\boldsymbol{c} \in \mathbf{R}^n$，$\boldsymbol{b} \in \mathbf{R}^m$. 令 p_j 是 A 的第 j 列，$j = 1,2,\cdots,n$，$\boldsymbol{x}^0$ 为(LP)的一个可行解. 令 $J = \{j \mid x_j^0 > 0\}$，$\bar{J} = \{j \mid x_j^0 = 0\}$. 则由线性规划的互补最优性条件知，$\bar{\boldsymbol{c}} \in F(\boldsymbol{x}^0)$，当且仅当下述问题有解：

$$\begin{cases} \boldsymbol{\pi} p_j = \bar{c}_j & j \in J, \\ \boldsymbol{\pi} p_j \leqslant \bar{c}_j & j \in \bar{J}, \end{cases} \tag{7-3}$$

其中 $\boldsymbol{\pi}^{\mathrm{T}} \in \mathbf{R}^m$.

若令 $\bar{c}_j = c_j + \theta_j - \alpha_j$，$j = 1,2,\cdots,n$，则(LP)的逆问题可写为

$$\begin{cases} \min \| \theta + \alpha \|, \\ \boldsymbol{\pi p}_j - \theta_j + \alpha_j = c_j, & j \in J, \\ \boldsymbol{\pi p}_j - \theta_j + \alpha_j \leqslant c_j, & j \in \bar{J}, \\ \theta_j, \alpha_j \geqslant 0, & j \in J \cup \bar{J}, \end{cases} \tag{7-4}$$

或者

$$\begin{cases} \min \| \theta + \alpha \|, \\ \boldsymbol{\pi p}_j - \theta_j + \alpha_j \leqslant c_j, & j \in J, \\ -\boldsymbol{\pi p}_j + \theta_j - \alpha_j \leqslant -c_j, & j \in J, \\ \boldsymbol{\pi p}_j - \theta_j + \alpha_j \leqslant c_j, & j \in \bar{J}, \\ \theta_j, \alpha_j \geqslant 0, & j \in J \cup \bar{J}. \end{cases} \tag{7-5}$$

显然,(7-5) 式的约束条件可写为

$$\begin{cases} \boldsymbol{\pi p}_j - \theta_j \leqslant c_j, & j \in J, \\ -\boldsymbol{\pi p}_j - \alpha_j \leqslant -c_j, & j \in J, \\ \boldsymbol{\pi p}_j - \theta_j \leqslant c_j, & j \in \bar{J}, \\ \theta_j \geqslant 0, & j \in J \cup \bar{J}, \\ \alpha_j \geqslant 0 & j \in J. \end{cases} \tag{7-6}$$

因此,在取模为 l_1 的情况下,(LP) 的逆问题为(ILP),它也是一个线性规划:

$$(\text{ILP1}) \quad \begin{cases} \min \sum_{j=1}^{n} \theta_j + \sum_{j \in J} \alpha_j, \\ \boldsymbol{\pi p}_j - \theta_j \leqslant c_j, & j \in J, \\ -\boldsymbol{\pi p}_j - \alpha_j \leqslant -c_j, & j \in J, \\ \boldsymbol{\pi p}_j - \theta_j \leqslant c_j, & j \in \bar{J}, \\ \theta_j \geqslant 0, & j \in J \cup \bar{J}, \\ \alpha_j \geqslant 0, & j \in J. \end{cases}$$

由线性规划的对偶性知,(ILP) 的对偶为

$$(\text{DILP1}) \quad \begin{cases} \min \{ \boldsymbol{c}^{\mathrm{T}} \boldsymbol{X} - \boldsymbol{c}_J^{\mathrm{T}} \boldsymbol{Y} \}, \\ \boldsymbol{AX} - \boldsymbol{A}_J \boldsymbol{Y} = 0, \\ \boldsymbol{0} \leqslant \boldsymbol{X} \leqslant \boldsymbol{1}, \\ \boldsymbol{0} \leqslant \boldsymbol{Y} \leqslant \boldsymbol{1}, \end{cases}$$

其中 $\boldsymbol{A}_J$ 是由 $\boldsymbol{A}$ 的所有使 $j \in J$ 的 P_j 组成,而 $\boldsymbol{c}_J$ 是由 $\boldsymbol{c}$ 中 $j \in J$ 的 c_j 组成的向量. $\boldsymbol{Y}$ 是 $|J|$ 维的列向量.

若在(7-5) 式中取 l_∞ 模时,则(LP) 的逆问题为

$$
\begin{cases}
\min u \\
\qquad \pi p_j - \theta_j \leqslant c_j, & j \in J, \\
\qquad -\pi pp_j - \alpha_j \leqslant -c_j, & j \in J, \\
\qquad \pi pp_j - \theta_j \leqslant c_j, & j \in \bar{J}, \\
\qquad \theta_j - u \leqslant 0, & j \in J \cup \bar{J}, \\
\qquad \alpha_j - u \leqslant 0, & j \in J, \\
\qquad \alpha_j, \theta_j \geqslant 0.
\end{cases} \tag{7-7}
$$

在(7-7)式中,把所有的 θ_j 和 α_j 都换为变量 u 后得

$$
(\text{ILP2}) \quad
\begin{cases}
\min u, \\
\qquad \pi p_j - u \leqslant c_j, & j \in \bar{J}, \\
\qquad -\pi p_j - u \leqslant -c_j, & j \in J, \\
\qquad \pi p_j - u \leqslant c_j, & j \in J,
\end{cases}
$$

可以证明(7-7)式和(ILP2)的最优值相等.事实上,若 $\{\pi,\theta,\alpha,u\}$ 是(7-7)式的任一可行解,则 $\{\pi,u\}$ 也是(ILP2)的可行解;反之,(ILP2)的任一个可行解 $\{\pi,u\}$,当 $u \geqslant 0$ 时,也可定义(7-7)式的可行解 $\{\pi,\theta,\alpha,u\}$,其中 $\theta_j = u, \alpha_j = u$.

(ILP2)的对偶为

$$
\begin{cases}
\min \sum_{j=1}^{n} c_j x_j - \sum_{j \in J} c_j y_j, \\
\qquad AX - A_J Y = 0,
\end{cases}
$$

$$
(\text{DILP2}) \quad
\begin{cases}
\sum_{j=1}^{n} x_j + \sum_{j \in J} y_j = 1, \\
x_j \geqslant 0, \quad j \in \bar{J} \cup J, \\
y_j \geqslant 0, \quad j \in J.
\end{cases}
$$

虽然有界变量的线性规划问题很容易变化为标准的线性规划模型(LP),但为了下面应用方便,还是具体地写出有界变量线性规划的逆问题的形式.

有界变量的线性规划为

$$
(\text{BLP}) \quad
\begin{cases}
\min c^{\mathrm{T}} X, \\
\qquad AX = b, \\
\qquad 0 \leqslant X \leqslant K,
\end{cases}
$$

(BLP)的对偶为

$$
(\text{DBLP}) \quad
\begin{cases}
\max \pi b - wK, \\
\qquad \pi A - w \leqslant c^{\mathrm{T}}, \\
\qquad w \geqslant 0.
\end{cases}
$$

设 X^0 是(BLP)的一个可行解,定义

$\bar{J}=\{j\mid x_j^0=0\}, J=\{j\mid x_j^0=k_j\}, J^0=\{j\mid 0<x_j^0<k_j\}$.

模仿(ILP1)的推导过程,在取范数 l_1 下,(BLP)的逆问题为

$$
(\text{IBLP1})\quad\begin{cases}\min \sum\limits_{j\in \bar{J}\cup J^0}\theta_j+\sum\limits_{j\in J^0\cup J}\alpha_j,\\ \pi p_j-\theta_j\leqslant c_j, & j\in\bar{J}\cup J^0,\\ -\pi p_j-\alpha_j\leqslant -c_j, & j\in J^0\cup J,\\ \theta_j\geqslant 0, & j\in\bar{J}\cup J^0,\\ \alpha_j\geqslant 0, & j\in J^0\cup J;\end{cases}
$$

(IBLP1)的对偶为

$$
(\text{DIBLP1})\quad\begin{cases}\min \sum\limits_{j\in \bar{J}\cup J^0}c_jx_j-\sum\limits_{j\in J^0\cup J}c_jy_j,\\ \sum\limits_{j\in \bar{J}\cup J^0}p_jx_j-\sum\limits_{j\in J^0\cup J}p_jy_j=0,\\ 0\leqslant x_j\leqslant 1,\\ 0\leqslant y_j\leqslant 1.\end{cases}
$$

类似地,在取范数为 l_∞ 下,(BLP)的逆问题为

$$
(\text{IBLP2})\quad\begin{cases}\min u,\\ \pi p_j-u\leqslant c_j, & j\in\bar{J}\cup J^0,\\ -\pi p_j-u\leqslant -c_j, & j\in J\cup J^0;\end{cases}
$$

(IBLP2)的对偶为

$$
(\text{DIBLP2})\quad\begin{cases}\min \sum\limits_{j\in \bar{J}\cup J^0}c_jx_j-\sum\limits_{j\in J\cup J^0}c_jy_j,\\ \sum\limits_{j\in \bar{J}\cup J^0}p_jx_j-\sum\limits_{j\in J^0\cup J}p_jy_j=0,\\ \sum\limits_{j\in \bar{J}\cup J^0}x_j+\sum\limits_{j\in J^0\cup J}y_j=1,\\ x_j\geqslant 0, & j\in\bar{J}\cup J^0,\\ y_j\geqslant 0, & j\in J^0\cup J.\end{cases}
$$

7.2 两类特殊逆线性规划的解

若问题(LP)有最优解 X^*,使 $\mathbf{0}\leqslant x^*\leqslant \mathbf{1}$,当(LP)的可行解 X^0 具有0或1分量时,则(DILP1)和(LP)的解之间有下述性质:

1° (DILP1)的任一个可行解$(x^{\mathrm{T}},y^{\mathrm{T}})$,若令

$$\tilde{x}_j=\begin{cases}x_j-y_j+x_j^0, & j\in J,\\ x_j, & j\in\bar{J}.\end{cases}$$

则 $\tilde{X}$ 是(LP)的可行解.

2° 设 X^* 是(LP)的最优解且 $0\leqslant X^*\leqslant 1$,则令

$$\hat{x}=x^*,\quad \hat{y}=x_j^0,$$

那么 $(\hat{x}^{\mathrm{T}},\hat{y}^{\mathrm{T}})$ 是(DILP1)的最优解.

由2°可得求解(ILP1)的算法如下:

步1　求解(LP),设最优解为 X^*,从而可得(DLP)的最优解 π^*.这儿设 $x^*\leqslant 1$).

步2　对一切 j,令 $\theta_j^*=0$;对 $j\in J$,令 $\alpha_j^*=c_j-\pi^*p_j$.那么 $\{\pi^*,\theta^*,\alpha^*\}$ 是(ILP)的最优解.

另一种特殊情况是,若(BLP)中的上界向量 K 各分量均为1.并且可行解 X^0 的各分量也是0或1,此时上述性质1°和2°,对(BLP1)和(DIBLP1)均成立,因此可得(IBLP1)的下述求解方法:

步1　求

$$\begin{cases}\min c^{\mathrm{T}}x,\\ \quad Ax=b,\\ \quad 0\leqslant x\leqslant 1\end{cases}$$

的最优解.设最优解为 x^*.同时可得其对偶问题

$$\begin{cases}\max\sum_{i=1}^{m}b_i\pi_i-\sum_{j=1}^{n}w_j,\\ \quad \pi A-w\leqslant c^{\mathrm{T}},\\ \quad\quad w_j\geqslant 0,\quad j=1,2,\cdots,n\end{cases}$$

的最优解 π^* 和 w^*.

步2　定义

$$\theta_j^*=\begin{cases}w_j^*, & 若\ j\in\bar{J}\ 且\ x_j^*>0,\\ 0, & 其余,\end{cases}$$

$$\alpha_j^*=\begin{cases}c_j-\pi^*p_j, & 若\ j\in J\ 且\ x_j^*=0,\\ 0, & 其余.\end{cases}$$

那么 $\{\pi^*,\theta^*,\alpha^*\}$ 是(IBLP1)的最优解.

7.3　逆最短路、指派和最小割问题

设 $N=(V,A;c)$ 是一有向网络,P 是 N 中自 s 到 t 的一条路.所谓**逆最短路问题**是指通过调整弧上的权向量 c 为 $\bar{c}$,使在保证 P 是网络 $\bar{N}=(V,A;\bar{c})$ 中自 s 到

t 的最短路的情况下，有 $\| \bar{c} - c \|$ 尽量小.

由于最短路问题的数学模型为

$$
(\mathrm{SP})\quad \begin{cases} \min \sum\limits_{(i,j)\in A} c_{ij}x_{ij} \\ -\sum\limits_{j\in\alpha(i)} x_{ij} + \sum\limits_{j\in\beta(i)} x_{ji} = \begin{cases} -1, & i = s, \\ 0, & i \neq s, t, \\ 1, & i = t, \end{cases} \\ x_{ij} \geq 0. \end{cases}
$$

定义(SP) 的可行解 x^0：

$$
x_{ij}^0 = \begin{cases} 1, & 若(i,j) \in P, \\ 0, & 其余. \end{cases}
$$

因为(SP) 中约束条件的系数矩阵为 N 的点 - 弧关联矩阵，因此该矩阵具有全单位模性质. 所以(SP) 具有最优解 x^*，使其分量为 0 或 1. 故可用 7.2 节中第一种方法求解.

指派问题可描述为下面的线性规划：

$$
(\mathrm{AP})\quad \begin{cases} \min \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} c_{ij}x_{ij}, \\ \sum\limits_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\cdots,n, \\ \sum\limits_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\cdots,n, \\ x_{ij} \geq 0. \end{cases}
$$

(AP) 的系数矩阵可视为等二部完全图的相邻矩阵，它也具有全单位模性质，因此它有整数最优解 x^*. 给定一个指派，等价于给定(AP) 一个整数可行解 x^0. 从而它也可用上述算法求其逆问题的解.

网络 $N = (V, A; c)$ 中分离 s 和 t 的最小割，可用下述规划表示：

$$
(\mathrm{MC})\quad \begin{cases} \min \sum\limits_{(ij)\in A} c_{ij}x_{ij}, \\ u_j - u_i + x_{ij} \geq 0, \quad (i,j) \in A, \\ u_s - u_t = 1, \\ x_{ij} \geq 0. \end{cases}
$$

容易看出(MC) 的系数矩阵是 N 的关联矩阵的转置和一个单位矩阵组合而成，它也是全单位模的. 它有 0,1- 整数最优解 (x^*, u^*). 给定一个 s-t 割 $(M, \bar{M})$，$S \in M, t \in \bar{M}$，令

$$
x_{ij}^0 = \begin{cases} 1, & (i,j) \in (M, \bar{M}), \\ 0, & 其余, \end{cases}
$$

$$u_i^0 = \begin{cases} 1, & i \in M, \\ 0, & i \in \bar{M}, \end{cases}$$

则(x^0, u^0)是(MC)的可行解.从而也可用上节方法求解逆最小 s-t 割问题.

具体地说,在 N 中去掉$(\bar{M}, M)$中的弧,然后求 $N = (V, A - (\bar{M}, M); c)$ 中自 s 到 t 的最大流.设它为$\{f^*(i,j)\}$.定义

$$\bar{c}_{(ij)} = \begin{cases} c_{ij} - f^*(i,j), & (i,j) \in (M, \bar{M}), \\ c_{ij}, & (i,j) \in A - (M, \bar{M}), \end{cases}$$

则$(M, \bar{M})$是网络 $N = (V, A; \bar{c})$ 的最小 s-t 割.

7.4 逆最小费用流问题

设 $N = (V, A; k, c)$,最小费用流问题可表示为

$$(\text{MCF}) \quad \begin{cases} \min \sum\limits_{(i,j)\in A} c_{ij} x_{ij}, \\ -\sum\limits_{j\in\alpha(i)} x_{ij} - \sum\limits_{j\in\beta(i)} x_{ji} = b_i, & i = 1, 2, \cdots, n, \\ 0 \leqslant x_{ij} \leqslant k_{ij}, & (i,j) \in A. \end{cases}$$

其中$\sum\limits_{i=1}^{n} b_i = 0$.若 $b_i > 0$,则 i 称为收点,若 $b_i < 0$,则 i 称为发点.

显然,约束方程的系数矩阵是 N 的点-弧关联矩阵乘以 -1.给定一个可行流 x^0.(MCF)的逆问题的对偶恰是(DIBLP1).

设 $N(x^0) = (V, \tilde{A}; 1, \tilde{c})$ 为 N 关于流 x^0 的剩余网络.其中 $\tilde{A} = \tilde{A}_1 \cup \tilde{A}_2$,而

$$\tilde{A}_1 = \{(i,j) \mid (i,j) \in A, x_{ij}^0 < k_{ij}\},$$

$$\tilde{A}_2 = \{(j,i) \mid (i,j) \in A, x_{ij}^0 > 0\},$$

$$\tilde{c}_{ij} = \begin{cases} c_{ij}, & (i,j) \in A_1, \\ -c_{ji}, & (i,j) \in A_2, \end{cases}$$

这儿1表示分量全是1的向量.

这样一来(DIBLP1)的约束方程组的系数矩阵就是 $N(x^0)$ 的关联矩阵.(DIBLP1)就是求解 $N(x^0)$ 的单位容量的最小费用的循环流.

(MCF)在模 l_∞ 下的逆问题的对偶为(DIBLP2),它是求 $N(x_0)$ 网络的最小平均费用圈问题.

7.5 逆最小支撑树问题

设 $N = (V, E; w)$ 是连通无向网络.T_0 是 N 的一棵支撑树.由第4章4.2节的

性质(1 和 2) 知，T_0 是 N 的最小权支撑树，当且仅当对每一条边 $e \in E \setminus T_0$，有 $w(e) \geqslant w(e')$ 对一切 $e' \in c(e)$ 成立. 因此，若 T_0 不是 N 的最小权支撑树，要使它在权 $\bar{w}$ 下是最小权支撑树，且使 $\| \bar{w} - w \|$ 最小，那么对一切 $e \in T_0$ 必有 $\bar{w}(e) \leqslant w(e)$；而 $e \in E \setminus T_0$，则必有 $\bar{w}(e) \geqslant w(e)$. 因此令

$$\bar{w}(e) = \begin{cases} w(e) - \theta(e), & e \in T_0, \\ w(e) + \alpha(e), & e \in E \setminus T_0, \end{cases}$$

于是关于 T_0 的逆最小支撑树问题可化为

$$(\text{IST}) \quad \begin{cases} \min \| \theta + \alpha \|, & \\ \theta(e') + \alpha(e) \geqslant w(e') - w(e), & e \in E \setminus T_0, e' \in c(e) \\ \theta(e) \geqslant 0, & e \in T_0, \\ \alpha(e) \geqslant 0, & e \in E \setminus T_0. \end{cases}$$

(IST) 在范数 l_1 下为(其中 $d(e', e) = w(e') - w(e)$)

$$(\text{IST1}) \quad \begin{cases} \min \sum_{e \in T_0} \theta(e) + \sum_{e \in E \setminus T_0} \alpha(e) & \\ \theta(e') + \alpha(e) \geqslant d(e', e), & e \in E \setminus T_0, e' \in c(e) \\ \theta(e), & \theta(e), \alpha(e) \geqslant 0. \end{cases}$$

(IST1) 的对偶为

$$(\text{DIST1}) \quad \begin{cases} \max \sum_{e' \in c(e)} \sum_{e \in E \setminus T_0} d(e', e) x(e', e), & \\ \sum_{e' \in c(e)} x(e'e) \leqslant 1, & e \in E \setminus T_0, \\ \sum_{e \in \Omega(e')} x(e', e) \leqslant 1, & e \in T_0, \\ x(e', e) \geqslant 0. & \end{cases}$$

这是二部图的最大权匹配问题，故可以用最小费用流问题求解.

有向图上的最小树形图的逆问题，以及划分限制的最小支撑树的逆问题，它们的对偶问题都可归结为最小费用流问题. 由于这些推导过于复杂，在此省略. 有兴趣读者参见有关文献.

参 考 文 献

1 Ahuja R K, Magnanti T L, Orlin J. B. Network flows. Englewwood cliffs: Printice-Hall, 1993.

2 Ford L R, Fulkerson D R. Flows in networks. Princeton: Princeton University Press, 1962

3 Liu Zhenhong. A solution to one of Kano's Conjecture Concerning kth Maximal Spanning Trees. ACTA Mathematicae Applicatae Sinica, 1988(4): 266 ~ 268

4 Lawler E. Combinalorial optimization: networks and matroids. New York: Holt Rinehart Winston, 1976.

5　Zhang Jianzhong, Liu Zhenhong, Calculating some inverse linear programming problems. J of computational and applied mathematics, 1996(72): 261 ~ 273.
6　刘振宏等. 具有次限制的最小树问题. 应用数学学报, 1980(3): 1 ~ 12.

·计算机数学卷·

第10篇

电路网络

编　者　卢华明
审校者　刘振宏

目　录

引　言

图论研究的问题有的可追溯到18世纪的数学家欧拉(Euler),近来它在计算机科学迅猛发展的刺激下,取得了引人瞩目的成就.在电路网络的研究方面尤为突出,它几乎改变了这个学科的面貌.随着科学技术的进步,电路的结构日益复杂,它的设计与分析更多地依赖于快速电子计算机;比如在计算机上作数字模拟来代替实验,图论在这里起到了重要的作用.下面的讨论将可看到,只要向计算机提供网络结构,计算机便可列出方程,并给出解,甚至于作出稳定性判断.这是计算机辅助设计最早、也是最成功的范例之一.在这一篇里集中介绍两个问题,一是"电路网络",另一是"信号流图".为了读者方便起见,简单介绍必要的图论基本概念.

1　电路网络图

1.1　广义的基尔霍夫定律

1.1.1　若干基本概念

假定电路的网络图用 $G=(V,E)$ 来表示,其中 $V=\{v_1,v_2,\cdots,v_n\}$ 是图 G 的顶点集合,$E=\{e_1,e_2,\cdots,e_m\}$ 表图 G 的边的集合.n 是顶点的数目,即 $|V|=n$;m 是边的数目,即 $|E|=m$.

为了避免和电学中的电动势等符号发生混淆,在以后的讨论中避免用 $v_1,v_2,\cdots,v_n$,和 $e_1,e_2,\cdots,e_m$ 等符号,一律用 $V=\{1,2,\cdots,n\}$,$E=\{1',2',\cdots,m'\}$ 来表示,这作为约定.

对于有向图 G,令

$$B=[b_{ij}]_{m\times n},$$

其中

$$b_{ij}=\begin{cases}0, & 若\ i\ 点与\ j'\ 边不关联,\\ 1, & 若\ i\ 点是\ j'\ 边的始点,\\ -1, & 若\ i\ 点是\ j'\ 边的终点.\end{cases}$$

B 称为图 G 的**关联矩阵**.

可见矩阵 B 的每一列都有且仅有两个非零元素:1 和 -1.

不特别申明都假定图 G 是**简单图**,即没有平行的边和自环(即始点和终点是同一点的边).

容易证明矩阵 B 的秩为 $n-1$,即只有 $n-1$ 行是独立的.从矩阵 B 中去掉与点 k 对应的行,余下的矩阵称为与 k 点对应的基本关联矩阵,用 $B_{(k)}$ 表示它.例如图

1-1有关联矩阵：

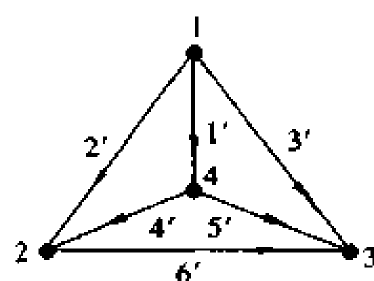


图 1-1

$$B=\begin{matrix}1\\2\\3\\4\end{matrix}\begin{bmatrix}1&1&1&0&0&0\\0&-1&0&-1&0&1\\0&0&-1&0&-1&-1\\-1&0&0&1&1&0\end{bmatrix},\quad (1'\ 2'\ 3'\ 4'\ 5'\ 6')$$

$$B_{(4)}=\begin{matrix}1\\2\\3\end{matrix}\begin{bmatrix}1&1&1&0&0&0\\0&-1&0&-1&0&1\\0&0&-1&0&-1&-1\end{bmatrix},\quad (1'\ 2'\ 3'\ 4'\ 5'\ 6')$$

$$B_{(1)}=\begin{matrix}2\\3\\4\end{matrix}\begin{bmatrix}0&-1&0&-1&0&1\\0&0&-1&0&-1&-1\\-1&0&0&1&1&0\end{bmatrix},\quad (1'\ 2'\ 3'\ 4'\ 5'\ 6')$$

n 个顶点的连通图 G 有一由 $n-1$ 条边构成的无回路的连通子图 T，T 称为图 G 的**支撑树**. 这 $n-1$ 条边称之为 T 的**树枝**. 余下的 $m-n+1$ 条边称为余树的边. 例如图 1-2 便是图 1-1 的支撑树；$1'$，$4'$ 和 $5'$ 便是树 T 的树枝.

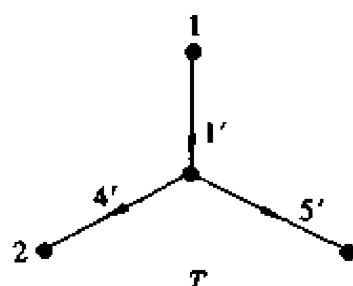


图 1-2

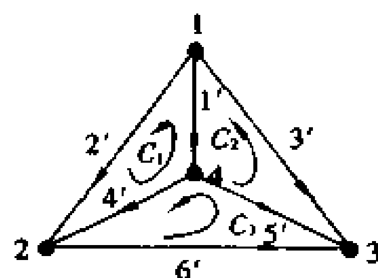


图 1-3

每一条属于余树的边可对应一回路，所以有 $m-n+1$ 个基本回路. 以图 1-1 和图 1-2 为例，可见有图 1-3 中的 C_1、C_2 和 C_3 三个基本回路，由余树边 $2'$，$3'$，$6'$ 所确定.

矩阵 $C=[c_{ij}]_{(m-n+1)\times m}$，其中

$$c_{ij}=\begin{cases}0, & \text{回路 } C_i \text{ 与边 } j' \text{ 无关联},\\ 1, & j' \text{ 边是回路 } C_i \text{ 上的边，且方向相同},\\ -1, & j' \text{ 边是回路 } C_i \text{ 上的边，但方向相反},\end{cases}$$

称为图 G 的**回路矩阵**.

例如图 1-3 的回路矩阵为

$$C=\begin{matrix}C_1\\C_2\\C_3\end{matrix}\begin{bmatrix}-1&1&0&-1&0&0\\1&0&-1&0&1&0\\0&0&0&1&-1&1\end{bmatrix}.\quad (1'\ 2'\ 3'\ 4'\ 5'\ 6')$$

已知连通图 $G=(V,E)$，S 是 E 的真子集．若从图 G 中消去属于 S 的所有边，则 G 图将被分解为非连通的两部分，但消去属于 S 的某一真子集的边，则图 G 仍保持为连通，这样的 S 称为图 G 的一个**割集**．割集实际上是使 G 失去其连通性的最小的边集合．

有向图 G 的割集 S 实际上将顶点分成不相交的两部分 V_1 和 V_2．从而可给 S 以方向，称之为有向割集．属于割集的边也可分成与 S 同向和反向两部分．

若 T 是图 G 的支撑树，j' 是 T 的树枝，消去 j' 则 T 失去连通性．可使树 T 的每一条树枝（设为 i'）对应一割集 S_i，S_i 不含其它属于 T 的边．因此对应于 $n-1$ 条树枝，有 $S_1,S_2,\cdots,S_{n-1}$ 一组基本割集．还是以图 1-1 和图 1-2 为例，有图 1-4 的割集 S_1、S_2、S_3．图中粗线为树 T，虚线为有向割集，箭头给出割集的方向．

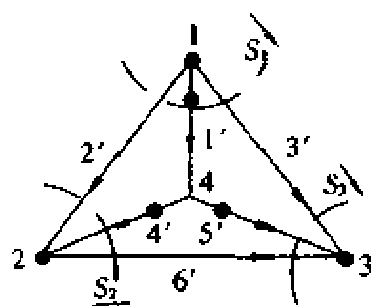


图 1-4

矩阵 $s=[s_{ij}]_{(n-1)\times m}$，其中

$$s_{ij}=\begin{cases}1, & \text{割集 } S_i \text{ 包含 } j' \text{ 边，且方向相同，}\\ -1, & \text{割集 } S_i \text{ 包含 } j' \text{ 边，但方向相反，}\\ 0, & \text{其它．}\end{cases}$$

称为图 G 的**割集矩阵**．

以图 1-4 为例，割集矩阵为

$$S=\begin{matrix} S_1 \\ S_2 \\ S_3 \\ \ \end{matrix}\begin{matrix}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \\ \begin{matrix} 1' & 2' & 3' & 4' & 5' & 6' \end{matrix}\end{matrix}.$$

重要的关系式：

若对回路矩阵中边的顺序作如下的安排：$m-n+1$ 条属于余树的边在前，而 $n-1$ 条属于树 T 的边排在后面，使得

$$C=[\underbrace{I}_{m-n+1} \ \vdots \ \underbrace{C_2}_{n-1}]\}m-n+1 \tag{1-1}$$

其中 I 是 $m-n-1$ 阶单位阵．相应地有

$$B_{(k)}=[\underbrace{B_1}_{m-n+1} \ \vdots \ \underbrace{B_2}_{n-1}]\}n-1 \tag{1-2}$$

$$S=[\underbrace{S_1}_{m-n+1} \ \vdots \ \underbrace{I}_{n-1}]\}n-1 \tag{1-3}$$

定理1 当边的顺序一致时，关联矩阵 B、回路矩阵 C、割集矩阵 S 有下列关系式成立：

1° $BC^{\mathrm{T}}=\mathbf{0}$，$CB^{\mathrm{T}}=\mathbf{0}$， (1-4)

2° $CS^{\mathrm{T}}=\mathbf{0}$，$SC^{\mathrm{T}}=\mathbf{0}$． (1-5)

其中 $\mathbf{0}$ 是零矩阵．

定理的结论是非常直观的，看一个实例便可明白，证明说的也就是这个道理．以图 1-5 为例，

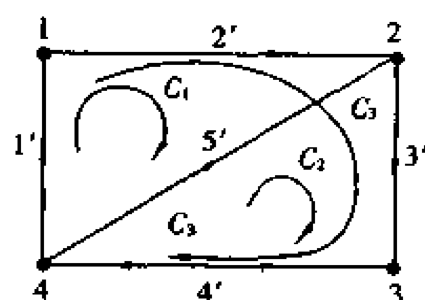


图 1-5

$$
B=\begin{matrix}1\\2\\3\\4\end{matrix}\begin{bmatrix}-1 & 1 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & -1\\ 0 & 0 & -1 & 1 & 0\\ 1 & 0 & 0 & -1 & 1\end{bmatrix},
\quad (\text{列: } 1'\ 2'\ 3'\ 4'\ 5')
$$

$$
C=\begin{matrix}C_1\\C_2\\C_3\end{matrix}\begin{bmatrix}1 & 1 & 0 & 0 & -1\\ 0 & 0 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 0\end{bmatrix},
\quad (\text{列: } 1'\ 2'\ 3'\ 4'\ 5')
$$

$$
BC^{\mathrm{T}}=\begin{bmatrix}-1 & 1 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & -1\\ 0 & 0 & -1 & 1 & 0\\ 1 & 0 & 0 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 1\\ 0 & 1 & 1\\ 0 & 1 & 1\\ -1 & 1 & 0\end{bmatrix}=\mathbf{0}
$$

以 B 矩阵第一行与 C^{T} 的第一列(即 C 的第一行)对应元素相乘时,发现 1′ 和 2′ 边与顶点 1 相关联,而且都在回路 C_1 上.但 1′ 边以 1 点为终点,故 $b_{11}=-1$,2′ 边以 1 点为始点,故 $b_{12}=1$,但 1′ 和 2′ 都与 C_1 的方向相同,即 $C_{11}=C_{12}=1$,故对应元素之积 $b_{11}C_{11}+b_{12}C_{12}=0$.

总之,当顶点 i 在 C_j 回路上时,必有两条边与 i 点相关联,它们在 C_j 回路上,和 C_j 的方向相同或相反,不妨假定它们与 C_j 方向相同.对于 i 点来说从一条边进入,从另一条边出去.关系式(1-4)说明的正是这个道理.

同样的道理,若 G 图的割集与某一回路有相同的边时,则相同边的边数必为偶数.不失一般性假定它们与回路方向一致,其中之一若与割集的方向一致,另一个必与割集的方向相反.关系式(1-5)成立便是这个道理.

定理 2　如若边的次序是余树边在前,树枝在后,使得 C、$B_{(k)}$、S 如(1-1)、(1-2)、(1-3)式所示,

$$
C=[\underbrace{I}_{m-n+1}\ \vdots\ \underbrace{C_2}_{n-1}]\ \}m-n+1
$$

$$
B_{(k)}=[\underbrace{B_1}_{m-n+1}\ \vdots\ \underbrace{B_2}_{n-1}]\ \}n-1
$$

$$
S=[\underbrace{S_1}_{m-n+1}\ \vdots\ \underbrace{I}_{n-1}]\ \}n-1
$$

则有

1° $C_2=-B_1^{\mathrm{T}}[B_2^{-1}]^{\mathrm{T}}=-B_1^{\mathrm{T}}[B_2^{\mathrm{T}}]^{-1}$.

2° $S_1=-C_2^{\mathrm{T}}=B_2^{-1}B_1$.

证　因为 $BC^{\mathrm{T}}=\mathbf{0}$,所以

$$
[B_1\ \vdots\ B_2]\begin{bmatrix}I\\ C_2^{\mathrm{T}}\end{bmatrix}=B_1+B_2C_2^{\mathrm{T}}=\mathbf{0},
$$

$$C_2^{\mathrm{T}} = -B_2^{-1}B_1, C_2 = -B^{\mathrm{T}}(B_2^{-1})^{\mathrm{T}} = -B_1^{\mathrm{T}}(B_2^{\mathrm{T}})^{-1}.$$

同理从 $SC^{\mathrm{T}} = 0$,可得

$$[S_1 \vdots I_{(n-1)}]\begin{bmatrix} I_{(m-n+1)} \\ C_2^{\mathrm{T}} \end{bmatrix} = S_1 + C_2^{\mathrm{T}} = 0,$$

所以
$$S_1 = -C_2^{\mathrm{T}}.$$

例 1 如图 1-6 所示，G 图的支撑树树枝为 1′,4′,5′.

$$B_{(4)} = \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}\left[\begin{array}{ccc:ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 \end{array}\right],$$
$$\quad 2' \quad 3' \quad 6' \quad 1' \quad 4' \quad 5'$$

$$B_1 = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

图 1-6

根据公式

$$C_2 = -B_1^{\mathrm{T}}[B_2^{-1}]^{\mathrm{T}} = -B_1^{\mathrm{T}}[B_2^{\mathrm{T}}]^{-1}.$$

因为

$$B_2 = B_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad [B_2^{-1}]^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

所以

$$C_2 = -\begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}.$$

所以

$$C = [\underbrace{I}_{m-n+1} \vdots \underbrace{C_2}_{n-1}] \} m-n+1$$

$$= \begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix}\left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 & -1 \end{array}\right].$$
$$\quad 2' \quad 3' \quad 6' \quad 1' \quad 4' \quad 5'$$

从图 1-6 可知结果正确,同样有

$$S = [\underbrace{S_1}_{m-n+1} \vdots \underbrace{I}_{n-1}] \} n-1$$

$$S_1 = -C_2^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix},$$

所以

$$S = \begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix}\left[\begin{array}{ccc:ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$
$$\quad\quad 2' \quad 3' \quad 6' \quad 1' \quad 4' \quad 5'$$

1.1.2　电路基本公式

电路网络图 $G = (V, E)$ 上各边的电流和电压降分别用

$$\boldsymbol{I} = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_m \end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

表示.

(1) 基尔霍夫电流定律:对于每一节点,流入该点的电流的代数和为零,即

$$\sum_{k=1}^{m} b_{jk} i_k(t) = 0, \quad j = 1, 2, \cdots, n.$$

或用矩阵表示为

$$\boldsymbol{BI} = \boldsymbol{0}$$

其中 $\boldsymbol{B}$ 是电路网络图 G 的关联矩阵.

(2) 基尔霍夫电压降定律:沿任回路 C,电压降的代数和为零,即

$$\sum_{k=1}^{m} c_{jk} v_k(t) = 0, \quad j = 1, 2, \cdots, m - n + 1.$$

或用矩阵表示为

$$\boldsymbol{CV} = \boldsymbol{0}.$$

(3) LRC 电路. 由基尔霍夫定律知,沿任一回路一圈,其电压降的总和为零. 假定回路上的电气元件是由线圈 L 电阻 R 及电容 C 组成. 设通过线圈 L 的电压降为 u_L,通过电阻 R 的电压降为 u_R,通过电容 C 的电压降为 u_C,则

$$u_L = L\frac{\mathrm{d}i}{\mathrm{d}t},$$

$$u_C = \frac{1}{C}\int_0^t i\mathrm{d}t,$$

$$u_R = Ri.$$

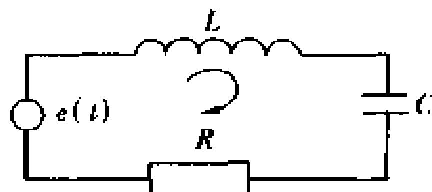


图 1-7

对于如图 1-7 由线圈 L、电阻 R、电容 C 及外加电压 $e(t)$ 构成的电路,根据基尔霍夫定律有

$$L\frac{\mathrm{d}i}{\mathrm{d}t} + \frac{1}{C}\int_0^t i\mathrm{d}t + Ri = e(t). \tag{1-6}$$

另有初始条件: $i|_{t=0} = i_0$.

(1-6) 式是一微分积分方程.

(1-6) 式两端对 t 求导,可得

$$L\frac{d^2 i}{dt^2}+R\frac{di}{dt}+\frac{1}{C}i=e'(t),$$

附加初始条件：

$$i\Big|_{t=0}=i_0,\quad \frac{di}{dt}\Big|_{t=0}=v_0,$$

问题导致二阶常微分方程初值问题.

若引进拉普拉斯变换，设 $i(t)$ 的拉普拉斯变换为 $I(p)$，即

$$\mathrm{L}\{i(t)\}=I(p).$$

根据拉普拉斯变换法则

$$\mathrm{L}\{\int_0^t i\mathrm{d}t\}=\frac{1}{p}I(p)-i_0,$$

$$\mathrm{L}\{\frac{\mathrm{d}i}{\mathrm{d}t}\}=pI(p).$$

令 $\mathrm{L}\{e(t)\}=E(p)$，对(1-6)式两端求拉普拉斯变换，得

$$(Lp+R+\frac{1}{Cp})I(p)=E(p)+i_0.$$

令 $Z(p)=Lp+R+\dfrac{1}{Cp}$，当 $i_0=0$ 时有

$$E(p)=Z(p)I(p),\tag{1-7}$$

称 $Z(p)$ 为**运算阻抗**，$I(p)$ 为**运算电流**，$E(p)$ 为**运算电压**，公式(1-7)称为**广义欧姆定律**，即运算电压等于运算阻抗乘以运算电流.

不仅如此，还可以证明，对如图 1-8、图 1-9 所示电路，有

$$Z(p)=Z_1(p)+Z_2(p),$$

$$\frac{1}{Z(p)}=\frac{1}{Z_1(p)}+\frac{1}{Z_2(p)}.$$

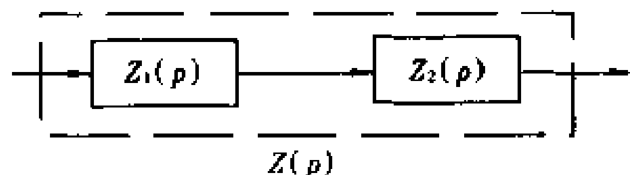


图 1-8

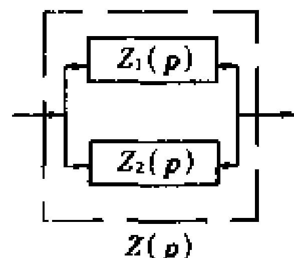


图 1-9

即电阻串联和并联的公式可推广到运算阻抗的串联和并联.

引进拉普拉斯变换后，电路的广义基尔霍夫定律及广义欧姆定律为

$$\begin{cases}\boldsymbol{B}_f\boldsymbol{I}(p)=\boldsymbol{0},\\ \boldsymbol{C}_f\boldsymbol{V}(p)=\boldsymbol{0},\\ \boldsymbol{V}(p)=\boldsymbol{Z}(p)\boldsymbol{I}(p).\end{cases}\tag{1-8}$$

这是关于 $\boldsymbol{I}(p)$、$\boldsymbol{V}(p)$ 的代数方程组.

矩阵 B_k 的秩为 $n-1$，矩阵 C_f 的秩为 $m-n+1$，$V(p)$ 为 m 维向量. 故方程组(1-8)包含 $2m$ 个独立的方程，可以求 $2m$ 个变量 $I(p)$ 和 $V(p)$ 的解. 解得 $I(p)$ 和 $V(p)$ 后，再求拉普拉斯逆变换. 电路问题可以从此得解.

理论分析如此，实际上很难照此办理. 电路稍为复杂一些，解方程组(1-8)得 $I(p)$ 和 $V(p)$ 及再求拉普拉斯逆变换都是十分困难的.

1.2 状态变量法

1.2.1 理论准备

后面的讨论都假定边的排列顺序为余树边在前，树枝边在后.

由 $B_kI=0$，可写成

$$[\underbrace{B_1}_{m-n+1} \; \vdots \; \underbrace{B_2}_{n-1}]\begin{bmatrix} I_{\bar{T}} \\ I_T \end{bmatrix}\begin{matrix} \}m-n+1 \\ \}n-1 \end{matrix} = 0,$$

其中 $I_{\bar{T}}$ 和 I_T 表示余树边和树枝边上的电流，所以

$$B_1I_{\bar{T}}+B_2I_T=0,$$

$$I_T=-B_2^{-1}B_1I_{\bar{T}}.$$

这公式表明 I_T 依赖于 $I_{\bar{T}}$，也就是说 $I_{\bar{T}}$ 一旦确定，I_T 便可通过上面公式计算出来.

同样由 $C_fV=0$ 可写成

$$[\underbrace{I}_{m-n+1} \; \vdots \; \underbrace{C_2}_{n-1}]\begin{bmatrix} V_{\bar{T}} \\ V_T \end{bmatrix}=0,$$

及定理 2 可得

$$V_{\bar{T}}=-C_2V_T=B_1^{\mathrm{T}}[B_2^{-1}]^{\mathrm{T}}V_T.$$

令 $K=B_1^{\mathrm{T}}[B_2^{-1}]^{\mathrm{T}}$，则

$$V_{\bar{T}}=KV_T,\quad I_T=-K^{\mathrm{T}}I_{\bar{T}},$$

或写成

$$\begin{bmatrix} V_{\bar{T}} \\ I_T \end{bmatrix}=\begin{bmatrix} 0 & K \\ -K^{\mathrm{T}} & 0 \end{bmatrix}\begin{bmatrix} I_{\bar{T}} \\ V_T \end{bmatrix}. \tag{1-9}$$

公式(1-9)在以后的讨论中扮演着十分重要的角色. 它说明电路网络上各边的电流和电压降并不是独立的变量，只要 $I_{\bar{T}}$ 和 V_T 求出来了，则 $V_{\bar{T}}$ 和 I_T 可以通过公式(1-9)求出来. 独立变量不再是 $2m$ 个. 不仅如此，还可以利用它列出关于 $I_{\bar{T}}$ 和 V_T 的微分方程，解得 $I_{\bar{T}}$ 和 V_T 后再一次利用来计算 I_T 和 $V_{\bar{T}}$，电路问题便得到解决. 先通过下面的例子来说明其原理. 然后导出一般方法.

例2 已知电路如图1-10(a)所示，(b)是对应的图 G，(c)是图 G 的支撑树(实线)和余树(虚线).

从图(c)可见树枝上包含了电容、电源及电阻，余树边上包含了电感及部分电阻.

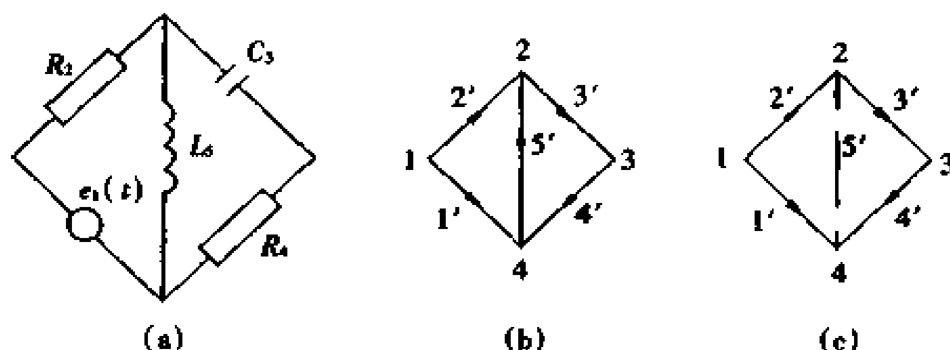


图 1-10

$$
B = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}\left[\begin{array}{cc:ccc} 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 \end{array}\right], \quad \begin{matrix} 5' & 4' & 3' & 1' & 2' \end{matrix}
$$

$$
B_{(4)} = \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}\left[\begin{array}{cc:ccc} 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \end{array}\right],
$$

所以

$$
B_1 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}.
$$

利用约当法求 B_2^{-1},得

$$
B_2^{-1} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix},
$$

于是得

$$
K^{\mathrm{T}} = B_2^{-1}B_1 = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix}.
$$

由 $V_{\bar{T}} = \begin{bmatrix} v_5 \\ v_4 \end{bmatrix}$, $I_T = \begin{bmatrix} i_3 \\ i_1 \\ i_2 \end{bmatrix}$,及$\begin{bmatrix} V_{\bar{T}} \\ I_T \end{bmatrix} = \begin{bmatrix} 0 & K \\ -K^{\mathrm{T}} & 0 \end{bmatrix}\begin{bmatrix} I_{\bar{T}} \\ V_T \end{bmatrix}$,得

$$
\begin{bmatrix} v_5 \\ v_4 \\ i_3 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} i_5 \\ i_4 \\ v_3 \\ e_1(t) \\ v_2 \end{bmatrix}. \tag{1-10}
$$

因5′、3′边包含电感、电容,故

$$v_5 = L_5 \frac{\mathrm{d}i_5}{\mathrm{d}t}, \quad i_3 = C_3 \frac{\mathrm{d}v_3}{\mathrm{d}t},$$

4′、2′边包含电阻,故

$$v_4 = R_4 i_4, \qquad i_2 = \frac{v_2}{R_2}.$$

将

$$V_T = \begin{bmatrix} v_5 \\ v_4 \end{bmatrix} = \begin{bmatrix} L_5 \frac{\mathrm{d}i_5}{\mathrm{d}t} \\ R_4 i_4 \end{bmatrix}, \quad I_T = \begin{bmatrix} i_3 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} C_3 \frac{\mathrm{d}v_3}{\mathrm{d}t} \\ i_1 \\ \frac{v_2}{R_2} \end{bmatrix}$$

代入(1-10)式得

$$\begin{bmatrix} L_5 \frac{\mathrm{d}i_5}{\mathrm{d}t} \\ R_4 i_4 \\ C_3 \frac{\mathrm{d}v_3}{\mathrm{d}t} \\ i_1 \\ \frac{v_2}{R_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_5 \\ i_4 \\ v_3 \\ e_1(t) \\ v_2 \end{bmatrix}, \tag{1-11}$$

展开得

$$\begin{cases} L_5 \dfrac{\mathrm{d}i_5}{\mathrm{d}A} = v_2 - e_1(t), & (1\text{-}12.\mathrm{a}) \\ C_3 \dfrac{\mathrm{d}v_3}{\mathrm{d}A} = i_4, & (1\text{-}12.\mathrm{b}) \\ R_4 i_4 = -v_2 - v_3 + e_1(t), & (1\text{-}12.\mathrm{c}) \\ \dfrac{v_2}{R_2} = i_4 - i_5, & (1\text{-}12.\mathrm{d}) \\ i_1 = -i_4 + i_5. & (1\text{-}12.\mathrm{e}) \end{cases}$$

从(1-12.a)—(1-12.e)式消去i_4和v_2导出关于v_3和i_5的微分方程,过程如下:

从(1-12.d)式得$v_2 = R_2(i_4 - i_5)$,从(1-12.c)式得$v_2 = -R_4 i_4 - v_3 + e_1(t)$,所以

$$v_2 = R_2(i_4 - i_5) = -R_4 i_4 - v_3 + e_1(t),$$

$$i_4 = \frac{R_2 i_5 - v_3 + e_1(t)}{R_2 + R_4}. \tag{1-13}$$

将它代入(1-12.b)式消去i_4得

$$\frac{\mathrm{d}v_3}{\mathrm{d}t} = \frac{R_2}{C_3(R_2 + R_4)} i_5 - \frac{1}{C_3(R_2 + R_4)} v_3 + \frac{1}{C_3(R_2 + R_4)} e_1(t). \tag{1-14}$$

从(1-12.c)及(1-12.b)得

$$
\begin{aligned}
v_2 &= -R_4 i_4 - v_3 + e_1(t) \\
&= \frac{-R_2R_4}{R_2+R_4} i_5 + \frac{R_4}{R_2+R_4} v_3 - \frac{R_4}{R_2+R_4} e_1(t) - v_3 + e_1(t) \\
&= \frac{-R_2R_4}{R_2+R_4} i_5 + \left(\frac{R_4}{R_2+R_4} - 1\right) v_3 - \left(\frac{R_4}{R_2+R_4} - 1\right) e_1(t) \\
&= \frac{-R_2R_4}{R_2+R_4} i_5 - \frac{R_2}{R_2+R_4} v_3 + \frac{R_2}{R_2+R_4} e_1(t),
\end{aligned}
$$

代入(1-12.a)式消去 v_2 得,

$$
\frac{\mathrm{d}i_5}{\mathrm{d}t} = \frac{-R_2R_4}{L_5(R_2+R_4)} i_5 - \frac{R_2}{L_5(R_2+R_4)} v_3 + \frac{R_2}{L_5(R_2+R_4)} e_1(t) - \frac{e_1(t)}{L_5}. \tag{1-15}
$$

(1-14)和(1-15)式联合得

$$
\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} i_5 \\ v_3 \end{bmatrix} = M\begin{bmatrix} i_5 \\ v_3 \end{bmatrix} + Ne_1(t),
$$

其中

$$
M = \begin{bmatrix} \dfrac{-R_2R_4}{L_5(R_2+R_4)} & \dfrac{-R_2}{L_5(R_2+R_4)} \\ \dfrac{R_2}{C_3(R_2+R_4)} & \dfrac{-1}{C_3(R_2+R_4)} \end{bmatrix}, \quad N = \begin{bmatrix} \dfrac{R_2}{L_5(R_2+R_4)} \\ \dfrac{1}{C_3(R_2+R_4)} \end{bmatrix}.
$$

1.2.2 状态变量法介绍

上节介绍的例子常有规律性,本节将它程式化,便于在计算机上操作.

第一步先从网络图中求一棵支撑树 T,假定它的树枝为含所有电容及电压源的边,补充以若干含电阻的边.这样的树若存在,称之为正常树(proper tree).自然,余下的余树边含有全部的电感,电流源及部分电阻.令

V_C、I_C 分别表示含电容边的电压及电流;

V_L、I_L 分别表示含电感边的电压及电流;

V_R、I_R 分别表示含电阻边的电压及电流;

V_G、I_G 分别表示含导纳边的电压及电流;

V_{CS}、I_{CS} 分别表示含电流源边的电压及电流;

V_{VS}、I_{VS} 分别表示含电压源边的电压及电流.

根据(1-9)式,

$$
\begin{bmatrix} V_T \\ I_{\bar{T}} \end{bmatrix} = \begin{bmatrix} 0 & K \\ -K^{\mathrm{T}} & 0 \end{bmatrix}\begin{bmatrix} I_T \\ V_{\bar{T}} \end{bmatrix}.
$$

由于 T 是正常树,树枝边含有所有的电容、电压源及部分电阻,余树边含所有的电感、电流源及部分电阻.故又可分成

$$V_{\bar{T}} = \begin{bmatrix} V_L \\ V_{CS} \\ V_R \end{bmatrix}, \qquad I_{\bar{T}} = \begin{bmatrix} I_L \\ I_{CS} \\ I_R \end{bmatrix},$$

$$V_T = \begin{bmatrix} V_C \\ V_{VS} \\ V_G \end{bmatrix}, \qquad I_T = \begin{bmatrix} I_C \\ I_{VS} \\ I_G \end{bmatrix}.$$

代入得

$$\begin{bmatrix} V_L \\ V_{CS} \\ V_R \\ I_C \\ I_{VS} \\ I_G \end{bmatrix} = \left[\begin{array}{ccc:ccc} & & & K_{11} & K_{12} & K_{13} \\ & 0 & & K_{21} & K_{22} & K_{23} \\ & & & K_{31} & K_{32} & K_{33} \\ \hdashline -K_{11}^T & -K_{21}^T & -K_{31}^T & & & \\ -K_{12}^T & -K_{22}^T & -K_{32}^T & & 0 & \\ -K_{13}^T & -K_{23}^T & -K_{33}^T & & & \end{array}\right] \begin{bmatrix} I_L \\ I_{CS} \\ I_R \\ V_C \\ V_{VS} \\ V_G \end{bmatrix}. \tag{1-16}$$

矩阵 K_{ij} 是矩阵 K 的分块子阵，$i,j = 1,2,3$. $i = 1,2,3$ 分别对应于 L,CS 和 R，$j = 1,2,3$ 分别对应于 C,VS,G. 以 $i = 1, j = 2$ 为例，K_{12} 为 $i = 1, j = 2$ 即 V_L 的线性表达式中关于 V_{VS} 的系数矩阵. 余此类推，不一一叙述. 由

$$V_L = L\frac{dI_L}{dt} = K_{11}V_C + K_{12}V_{CS} + K_{13}V_G,$$

$$I_C = C\frac{dV_C}{dt} = -K_{11}^T I_L - K_{21}^T I_{CS} - K_{31}^T I_R,$$

得

$$\begin{cases} \dfrac{dI_L}{dt} = L^{-1}K_{11}V_C + L^{-1}K_{12}V_{CS} + L^{-1}K_{13}V_C, \\ \dfrac{dV_C}{dt} = -C^{-1}K_{11}^T I_L - C^{-1}K_{21}^T I_{CS} - C^{-1}K_{31}^T I_R. \end{cases} \tag{1-17}$$

令

$$X = \begin{bmatrix} I_L \\ V_C \end{bmatrix}, \quad Y = \begin{bmatrix} I_{CS} \\ V_{VS} \end{bmatrix}, \quad W = \begin{bmatrix} I_R \\ V_G \end{bmatrix},$$

则

$$\frac{dX}{dt} = MX + NY + PW \tag{1-18}$$

其中

$$M = \begin{bmatrix} 0 & L^{-1}K_{11} \\ -C^{-1}K_{11}^T & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & L^{-1}K_{12} \\ -C^{-1}K_{21}^T & 0 \end{bmatrix},$$

$$P = \begin{bmatrix} 0 & L^{-1}K_{13} \\ -C^{-1}K_{31}^T & 0 \end{bmatrix}.$$

电流源的电流 I_{CS} 和电压源的电压 V_{VS} 是已知量，即 Y 是已知量.由

$$V_R = RI_R = K_{31}V_C + K_{32}V_{CS} + K_{33}V_G,$$

$$I_G = GV_G = -K_{13}^TI_L - K_{23}^TI_{CS} - K_{33}^TI_R,$$

或

$$\begin{bmatrix} I_R \\ V_G \end{bmatrix} = \begin{bmatrix} 0 & R^{-1}K_{31} \\ -G^{-1}K_{31}^T & 0 \end{bmatrix}\begin{bmatrix} I_L \\ V_C \end{bmatrix} + \begin{bmatrix} 0 & R^{-1}K_{32} \\ -G^{-1}K_{23}^T & 0 \end{bmatrix}\begin{bmatrix} I_{CS} \\ V_{VS} \end{bmatrix} + \begin{bmatrix} 0 & R^{-1}K_{33} \\ -G^{-1}K_{33}^T & 0 \end{bmatrix}\begin{bmatrix} I_R \\ V_G \end{bmatrix},$$

或

$$\begin{bmatrix} I_{11} & -R^{-1}K_{33} \\ G^{-1}K_{33}^T & I_{22} \end{bmatrix}\begin{bmatrix} I_R \\ V_G \end{bmatrix} = \begin{bmatrix} 0 & R^{-1}K_{31} \\ -G^{-1}K_{13}^T & 0 \end{bmatrix}\begin{bmatrix} I_L \\ V_C \end{bmatrix} + \begin{bmatrix} 0 & R^{-1}K_{32} \\ -G^{-1}K_{23}^T & 0 \end{bmatrix}\begin{bmatrix} I_{CS} \\ V_{VS} \end{bmatrix}$$

得

$$W = \begin{bmatrix} I_R \\ V_G \end{bmatrix} = DX + EY, \tag{1-19}$$

其中

$$X = \begin{bmatrix} I_L \\ V_C \end{bmatrix}, \quad Y = \begin{bmatrix} I_{CS} \\ V_{VS} \end{bmatrix},$$

$$D = \begin{bmatrix} I_{11} & -R^{-1}K_{33} \\ G^{-1}K_{33}^T & I_{22} \end{bmatrix}^{-1}\begin{bmatrix} 0 & R^{-1}K_{31} \\ -G^{-1}K_{13}^T & 0 \end{bmatrix},$$

$$E = \begin{bmatrix} I_{11} & -R^{-1}K_{33} \\ G^{-1}K_{33}^T & I_{22} \end{bmatrix}^{-1}\begin{bmatrix} 0 & R^{-1}K_{32} \\ -G^{-1}K_{23}^T & 0 \end{bmatrix}.$$

代入(1-18)式消去 W 得关于 X 的微分方程组：

$$\frac{dX}{dt} = \bar{A}X + \bar{B}Y, \tag{1-20}$$

其中

$$\bar{A} = \begin{bmatrix} 0 & L^{-1}K_{11} \\ -C^{-1}K_{11}^T & 0 \end{bmatrix} + \begin{bmatrix} 0 & L^{-1}K_{13} \\ -C^{-1}K_{31}^T & 0 \end{bmatrix}D,$$

$$\bar{B} = \begin{bmatrix} 0 & L^{-1}K_{12} \\ -C^{-1}K_{21}^T & 0 \end{bmatrix} + \begin{bmatrix} 0 & L^{-1}K_{13} \\ -C^{-1}K_{31}^T & 0 \end{bmatrix}E.$$

另一方面从(1-16)式不难得到

$$\begin{bmatrix} V_{CS} \\ I_{VS} \end{bmatrix} = \begin{bmatrix} 0 & K_{21} \\ -K_{12}^T & 0 \end{bmatrix}X + \begin{bmatrix} 0 & K_{22} \\ -K_{22}^T & 0 \end{bmatrix}Y + \begin{bmatrix} 0 & K_{23} \\ -K_{32}^T & 0 \end{bmatrix}W,$$

再将(1-19)式代入，消去 W 可得

$$\begin{bmatrix} V_{CS} \\ I_{cs} \end{bmatrix} = FX + RY. \tag{1-21}$$

(1-21) 式说明电源边的电压和电压源边电流可用 X 表示. 其中

$$F = \begin{bmatrix} 0 & K_{21} \\ -K_{12}^{T} & 0 \end{bmatrix} + \begin{bmatrix} 0 & K_{23} \\ -K_{32}^{T} & 0 \end{bmatrix} D,$$

$$R = \begin{bmatrix} 0 & K_{22} \\ -K_{22}^{T} & 0 \end{bmatrix} + \begin{bmatrix} 0 & K_{23} \\ -K_{32}^{T} & 0 \end{bmatrix} E.$$

解微分方程(1-20),得 X,代入(1-19)式得$\begin{bmatrix} I_{R} \\ V_{G} \end{bmatrix}$,再代入(1-21)式得$\begin{bmatrix} V_{CS} \\ I_{VS} \end{bmatrix}$.

称 X 为状态变量. 这样一个解题过程称之为状态变量法. 利用它可设计一软件,使得只要提供电路的网络结构及有关参数,计算机便可自动列出方程,并求解. 电路设计的效率大大提高,开了计算机辅助设计的先河. 不仅如此,这也是利用计算机算来代替实验的成功范例.

1.2.3　举例

例3　电路如图1-11(a)所示,求解此电路问题,其中图(b)是图(a)的网络图,图(c)是图(b)的树. 树包含了所有的电容,所以是正常树.

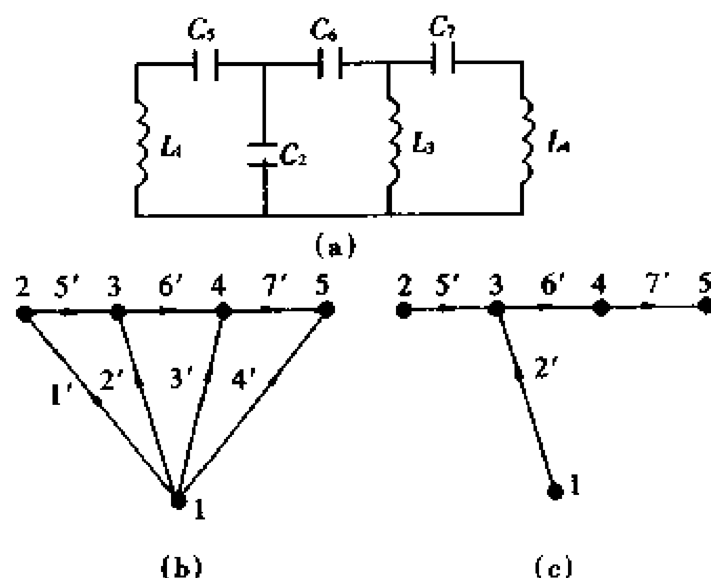


图 1-11

$$B = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \\ \end{array}\begin{array}{c} \left[\begin{array}{ccc:cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{array}\right] \\ \begin{array}{ccc:cccc} 1' & 3' & 4' & 2' & 5' & 6' & 7' \end{array} \end{array},$$

$$
\boldsymbol{B}_{(1)}=\begin{matrix}2\\3\\4\\5\end{matrix}\left[\begin{array}{ccc:cccc}-1&0&0&0&1&0&0\\0&0&0&-1&-1&1&0\\0&-1&0&0&0&-1&1\\0&0&-1&0&0&0&-1\end{array}\right],
$$
$$
\begin{array}{ccc:cccc}1'&3'&4'&2'&5'&6'&7'\end{array}
$$

$$
\boldsymbol{B}_1=\begin{bmatrix}-1&0&0\\0&0&0\\0&-1&0\\0&0&-1\end{bmatrix},\quad \boldsymbol{B}_2=\begin{bmatrix}0&1&0&0\\-1&-1&1&0\\0&0&-1&1\\0&0&0&-1\end{bmatrix},
$$

利用约当法求 $\boldsymbol{B}_2^{-1}$,得

$$
\boldsymbol{B}_2^{-1}=\begin{bmatrix}-1&-1&-1&-1\\1&0&0&0\\0&0&-1&-1\\0&0&0&-1\end{bmatrix}.
$$

于是有

$$
\boldsymbol{K}^{\mathrm{T}}=\boldsymbol{B}_2^{-1}\boldsymbol{B}_1=\begin{bmatrix}-1&-1&-1&-1\\1&0&0&0\\0&0&-1&-1\\0&0&0&-1\end{bmatrix}\begin{bmatrix}-1&0&0\\0&0&0\\0&-1&0\\0&0&-1\end{bmatrix}
$$
$$
=\begin{bmatrix}1&1&1\\-1&0&0\\0&1&1\\0&0&1\end{bmatrix}.
$$

$$
\boldsymbol{V}_{\bar{T}}=\begin{bmatrix}v_1\\v_3\\v_4\end{bmatrix}=\begin{bmatrix}L_1\dfrac{\mathrm{d}i_1}{\mathrm{d}t}\\L_3\dfrac{\mathrm{d}i_3}{\mathrm{d}t}\\L_4\dfrac{\mathrm{d}i_4}{\mathrm{d}t}\end{bmatrix},\quad \boldsymbol{I}_{\bar{T}}=\begin{bmatrix}i_1\\i_3\\i_4\end{bmatrix},
$$

$$
\boldsymbol{I}_T=\begin{bmatrix}i_2\\i_5\\i_6\\i_7\end{bmatrix}=\begin{bmatrix}C_2\dfrac{\mathrm{d}v_2}{\mathrm{d}t}\\C_5\dfrac{\mathrm{d}v_5}{\mathrm{d}t}\\C_6\dfrac{\mathrm{d}v_6}{\mathrm{d}t}\\C_7\dfrac{\mathrm{d}v_7}{\mathrm{d}t}\end{bmatrix},\quad \boldsymbol{V}_T=\begin{bmatrix}v_2\\v_5\\v_6\\v_7\end{bmatrix}.
$$

代入(1-9)式得

$$\begin{bmatrix} L_1\frac{di_1}{dt} \\ L_3\frac{di_3}{dt} \\ L_4\frac{di_4}{dt} \\ C_2\frac{dv_2}{dt} \\ C_5\frac{dv_5}{dt} \\ C_6\frac{dv_6}{dt} \\ C_7\frac{dv_7}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_1 \\ i_3 \\ i_4 \\ v_2 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix},$$

所以

$$\frac{dX}{dt} = AX,$$

$$X = (i_1, i_3, i_4, v_2, v_5, v_6, v_7)^T,$$

其中

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{L_1} & -\frac{1}{L_1} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{L_3} & 0 & \frac{1}{L_3} & 0 \\ 0 & 0 & 0 & \frac{1}{L_4} & 0 & \frac{1}{L_4} & \frac{1}{L_4} \\ -\frac{1}{C_2} & -\frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 & 0 & 0 \\ -\frac{1}{C_5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{C_4} & -\frac{1}{C_6} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{C_7} & 0 & 0 & 0 & 0 \end{bmatrix}.$$

例 4　已知电路为图1-12所示，解此电路.

图 1-12 电路的图和其支撑树分别如图 1-13 的(a)，(b) 所示. 图的基本割集矩阵

$$S = \begin{matrix} S_1 \\ S_2 \end{matrix}\begin{bmatrix} 1 & 0 & -1 & \vdots & 1 & 0 \\ 0 & 1 & 1 & \vdots & 0 & 1 \end{bmatrix},$$
$$\quad\quad 1' \quad 3' \quad 5' \quad 2' \quad 4'$$

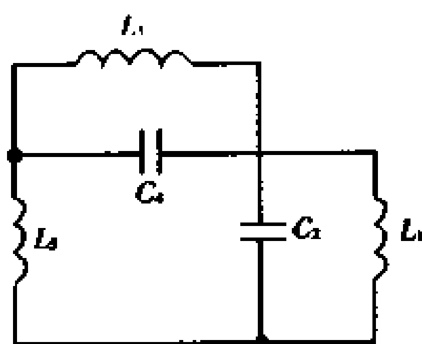


图 1-12

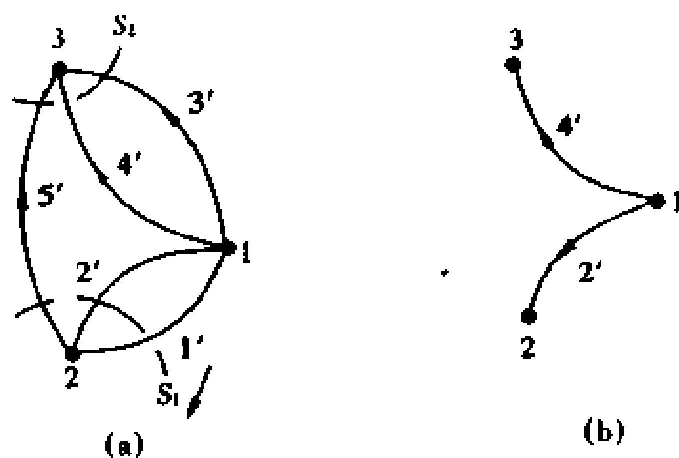


图 1-13

$$K = S_1^{\mathrm{T}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \end{bmatrix}.$$

令

$$V_{\bar{T}} = \begin{bmatrix} v_1 \\ v_3 \\ v_5 \end{bmatrix} = \begin{bmatrix} L_1 \dfrac{\mathrm{d}i_1}{\mathrm{d}t} \\ L_3 \dfrac{\mathrm{d}i_3}{\mathrm{d}t} \\ L_5 \dfrac{\mathrm{d}i_5}{\mathrm{d}t} \end{bmatrix}, \quad I_{\bar{T}} = \begin{bmatrix} i_1 \\ i_3 \\ i_5 \end{bmatrix},$$

$$I_T = \begin{bmatrix} i_2 \\ i_4 \end{bmatrix} = \begin{bmatrix} C_2 \dfrac{\mathrm{d}v_2}{\mathrm{d}t} \\ C_4 \dfrac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix}, \quad V_T = \begin{bmatrix} v_2 \\ v_4 \end{bmatrix},$$

所以

$$\begin{bmatrix} L_1\dfrac{\mathrm{d}i_1}{\mathrm{d}t} \\ L_3\dfrac{\mathrm{d}i_3}{\mathrm{d}t} \\ L_5\dfrac{\mathrm{d}i_5}{\mathrm{d}t} \\ C_2\dfrac{\mathrm{d}v_2}{\mathrm{d}t} \\ C_4\dfrac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_1 \\ i_3 \\ i_5 \\ v_2 \\ v_4 \end{bmatrix}.$$

或写成

$$\frac{\mathrm{d}X}{\mathrm{d}t} = AX,$$

其中

$$X = (i_1, i_3, i_5, v_2, v_4)^{\mathrm{T}}$$

$$A = \begin{bmatrix} \dfrac{1}{L_1} & 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{L_3} & 0 & 0 & 0 \\ 0 & 0 & \dfrac{1}{L_5} & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1}{C_2} & 0 \\ 0 & 0 & 0 & 0 & \dfrac{1}{C_4} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & \dfrac{1}{L_1} & 0 \\ 0 & 0 & 0 & 0 & \dfrac{1}{L_3} \\ 0 & 0 & 0 & -\dfrac{1}{L_5} & \dfrac{1}{L_5} \\ -\dfrac{1}{C_2} & 0 & \dfrac{1}{C_2} & 0 & 0 \\ 0 & -\dfrac{1}{C_4} & -\dfrac{1}{C_4} & 0 & 0 \end{bmatrix}.$$

例 5　电路如图1-14(a) 所示. 求解此电路.

图(b) 是图(a) 的网络图,图(c) 是图(b) 的支撑树,由于树枝上都是电容和电压源,所以是正常树. 令

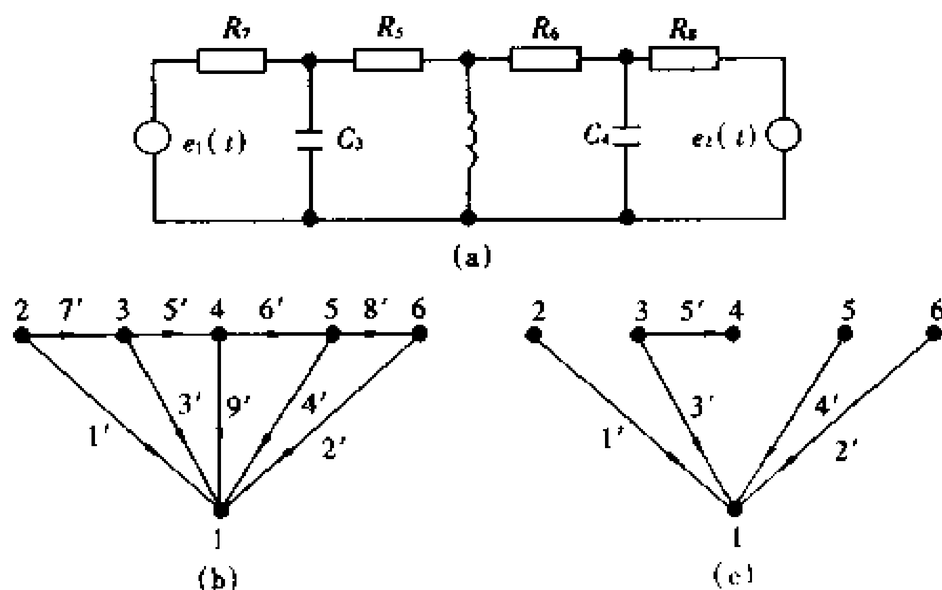


图 1-14

$$
\boldsymbol{V}_{\bar{T}}=\begin{bmatrix} v_9\\ v_6\\ v_7\\ v_8 \end{bmatrix}=\begin{bmatrix} L_9\dfrac{\mathrm{d}i_9}{\mathrm{d}t}\\ R_6 i_6\\ R_7 i_7\\ R_8 i_8 \end{bmatrix},\quad \boldsymbol{I}_{\bar{T}}=\begin{bmatrix} i_9\\ i_6\\ i_7\\ i_8 \end{bmatrix},
$$

$$
\boldsymbol{I}_T=\begin{bmatrix} i_3\\ i_4\\ i_1\\ i_2\\ i_5 \end{bmatrix}=\begin{bmatrix} C_3\dfrac{\mathrm{d}v_3}{\mathrm{d}t}\\ C_4\dfrac{\mathrm{d}v_4}{\mathrm{d}t}\\ i_1\\ i_2\\ \dfrac{v_5}{R_5} \end{bmatrix},\quad \boldsymbol{V}_T=\begin{bmatrix} v_3\\ v_4\\ e_1(t)\\ e_2(t)\\ v_5 \end{bmatrix}.
$$

图(b)的关联矩阵

$$
\boldsymbol{B}=\begin{array}{c} 1\\2\\3\\4\\5\\6\\ \\ \end{array}\begin{array}{c}\left[\begin{array}{cccc:ccccc} 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1\\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right]\\ \begin{array}{cccc:ccccc} 9' & 6' & 7' & 8' & 3' & 4' & 1' & 2' & 5' \end{array}\end{array},
$$

$$
B_{(1)} = \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}\left[\begin{array}{cccc:ccccc}
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
\hline
9' & 6' & 7' & 8' & 3' & 4' & 1' & 2' & 5'
\end{array}\right],
$$

于是

$$
B_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
B_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},
$$

利用约当法求得

$$
B_2^{-1} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix},
$$

于是有

$$
\begin{aligned}
K &= B_1^{\mathrm{T}}(B_2^{-1})^{\mathrm{T}} \\
&= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix},
\end{aligned}
$$

代入公式

$$
\begin{bmatrix} V_T \\ I_T \end{bmatrix} = \begin{bmatrix} 0 & K \\ -K^T & 0 \end{bmatrix}\begin{bmatrix} I_T \\ V_T \end{bmatrix},
$$

得

$$
\begin{bmatrix} L_9\dfrac{\mathrm{d}i_9}{\mathrm{d}t} \\ R_6 i_6 \\ R_7 i_7 \\ R_8 i_8 \\ C_3\dfrac{\mathrm{d}v_3}{\mathrm{d}t} \\ C_4\dfrac{\mathrm{d}v_4}{\mathrm{d}t} \\ i_1 \\ i_2 \\ \dfrac{v_5}{R_5} \end{bmatrix}
=
\left[\begin{array}{cccc|ccccc}
 & & & & -1 & 0 & 0 & 0 & 1 \\
 & \multicolumn{2}{c}{\mathbf{0}} & & -1 & 1 & 0 & 0 & 1 \\
 & & & & -1 & 0 & 1 & 0 & 0 \\
 & & & & 0 & -1 & 0 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & & & & & \\
0 & -1 & 0 & 1 & & & & & \\
0 & 0 & -1 & 0 & & & \mathbf{0} & & \\
0 & 0 & 0 & -1 & & & & & \\
-1 & -1 & 0 & 0 & & & & &
\end{array}\right]
\begin{bmatrix} i_9 \\ i_6 \\ i_7 \\ i_8 \\ v_3 \\ v_4 \\ e_1(t) \\ e_2(t) \\ v_5 \end{bmatrix},
$$

即得

$$L_9\frac{\mathrm{d}i_9}{\mathrm{d}t} = v_5 - v_3, \tag{1-22}$$

$$C_3\frac{\mathrm{d}v_3}{\mathrm{d}t} = i_6 + i_7 + i_9, \tag{1-23}$$

$$C_4\frac{\mathrm{d}v_4}{\mathrm{d}t} = i_8 - i_6, \tag{1-24}$$

$$v_5 = -R_5(i_6 + i_9), \tag{1-25}$$

$$i_6 = \frac{v_4 - v_3 + v_5}{R_6}, \tag{1-26}$$

$$i_7 = \frac{-v_3 + e_1(t)}{R_7}, \tag{1-27}$$

$$i_8 = \frac{e_2(t) - v_4}{R_8}, \tag{1-28}$$

$$i_1 = -i_7, \tag{1-29}$$

$$i_2 = -i_8. \tag{1-30}$$

从(1-25)式可得

$$v_5 + R_5 i_6 = -R_5 i_9, \tag{1-31}$$

从(1-26)式可得

$$\frac{-v_5}{R_6} + i_6 = \frac{v_4 - v_3}{R_6}; \tag{1-32}$$

(1-31)和(1-32)式联立求解可得

$$
v_5=\frac{\begin{vmatrix}-R_5 i_9 & R_5\\ \dfrac{v_4-v_3}{R_6} & 1\end{vmatrix}}{\begin{vmatrix}1 & R_5\\ -\dfrac{1}{R_6} & 1\end{vmatrix}}=\frac{-R_5 i_9+(v_3-v_4)\dfrac{R_5}{R_6}}{1+\dfrac{R_5}{R_6}}
$$

$$
=\frac{R_5 v_3-R_5 v_4-R_5 R_6 i_9}{R_5+R_6}, \tag{1-33}
$$

$$
v_6=\frac{\begin{vmatrix}1 & -R_5 i_9\\ -\dfrac{1}{R_6} & \dfrac{v_4-v_3}{R_6}\end{vmatrix}}{\left(1+\dfrac{R_5}{R_6}\right)}=\frac{(v_4-v_3)-R_5 i_9}{R_5+R_6}. \tag{1-34}
$$

代人(1-22)式得

$$
L_9\frac{\mathrm{d}i_9}{\mathrm{d}t}=v_5-v_3=\frac{R_5 v_3-R_5 R_6 i_9-R_5 v_4}{R_5+R_6}-v_3
$$

$$
=\frac{-R_6}{R_5+R_6}v_3-\frac{R_5}{R_5+R_6}v_4-\frac{R_5 R_6}{R_5+R_6}i_9.
$$

同理可得

$$
C_3\frac{\mathrm{d}v_3}{\mathrm{d}t}=i_6+i_7+i_9
$$

$$
=\frac{1}{R_5+R_6}v_4-\left(\frac{1}{R_5+R_6}+\frac{1}{R_7}\right)v_3+\frac{R_6}{R_5+R_6}i_9+\frac{1}{R_7}e_1(t),
$$

$$
C_4\frac{\mathrm{d}v_4}{\mathrm{d}t}=\frac{1}{R_5+R_6}v_3-\left(\frac{1}{R_5+R_6}+\frac{1}{R_8}\right)v_4+\frac{R_5}{R_5+R_6}i_9+\frac{1}{R_8}e_2(t).
$$

写成矩阵形式有

$$
\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}i_9\\ v_3\\ v_4\end{bmatrix}=\begin{bmatrix}\dfrac{-R_5R_6}{L_9(R_5+R_6)} & \dfrac{-R_6}{L_9(R_5+R_6)} & \dfrac{-R_5}{L_9(R_5+R_6)}\\ \dfrac{R_6}{C_3(R_5+R_6)} & \dfrac{-R_5-R_6-R_7}{C_3(R_5+R_6)R_7} & \dfrac{1}{C_3(R_5+R_6)}\\ \dfrac{R_5}{C_4(R_5+R_6)} & \dfrac{1}{C_4(R_5+R_6)} & \dfrac{-R_5-R_6-R_7}{C_4(R_5+R_6)R_8}\end{bmatrix}\begin{bmatrix}i_9\\ v_3\\ v_4\end{bmatrix}+
$$

$$
\begin{bmatrix}0 & 0\\ \dfrac{1}{C_3R_7} & 0\\ 0 & \dfrac{1}{C_4R_8}\end{bmatrix}\begin{bmatrix}e_1(t)\\ e_2(t)\end{bmatrix}.
$$

下面讨论如何利用公式(1-17)和(1-20).本例

$\boldsymbol{V}_{\mathrm{L}}=(v_9)$,　$\boldsymbol{V}_{\mathrm{R}}=(v_6,v_7,v_8)^{\mathrm{T}}$,　$\boldsymbol{V}_{\mathrm{C}}=(v_3,v_4)$,　$\boldsymbol{I}_{\mathrm{C}}=(i_3,i_4)^{\mathrm{T}}$;

$$K=\left[\begin{array}{cc:cc:c}-1&0&0&0&1\\ \hdashline -1&1&0&0&1\\ -1&0&1&0&0\\ 0&-1&0&1&0\end{array}\right],$$

$$K_{11}=[-1\quad 0],\quad K_{12}=[0\quad 0],\quad K_{13}=[1],$$

$$K_{31}=\begin{bmatrix}-1&1\\-1&0\\0&-1\end{bmatrix},\quad K_{32}=\begin{bmatrix}0&0\\1&0\\0&1\end{bmatrix},\quad K_{33}=\begin{bmatrix}1\\0\\0\end{bmatrix},$$

由于本例无电流源,故 K_{21},K_{22},K_{23} 为空.

$$L=[L_9],\quad C=\mathrm{diag}(C_3,C_4)=\begin{bmatrix}C_3&0\\0&C_4\end{bmatrix}.$$

直接代入(1-17)式,其中

$$X=(i_9,v_3,v_4)^{\mathrm T},\quad Y=(e_1(t),e_2(t))^{\mathrm T},\quad W=(i_6,i_7,i_8,v_5)^{\mathrm T},$$

$$L^{-1}K_{11}=[\frac{1}{L_9}][-1\quad 0]=[-\frac{1}{L_9}\quad 0],\quad K_{12}=[0\quad 0],$$

$$-C^{-1}K_{11}^{\mathrm T}=\begin{bmatrix}-\frac{1}{C_3}&0\\0&-\frac{1}{C_4}\end{bmatrix}\begin{bmatrix}-1\\0\end{bmatrix}=\begin{bmatrix}\frac{1}{C_3}\\0\end{bmatrix},$$

$$L^{-1}K_{13}=[\frac{1}{L_9}],$$

$$-C^{-1}K_{31}^{\mathrm T}=\begin{bmatrix}-\frac{1}{C_3}&0\\0&-\frac{1}{C_4}\end{bmatrix}\begin{bmatrix}-1&-1&0\\1&0&-1\end{bmatrix}=\begin{bmatrix}\frac{1}{C_3}&\frac{1}{C_3}&0\\-\frac{1}{C_4}&0&\frac{1}{C_4}\end{bmatrix}.$$

得

$$\frac{\mathrm d}{\mathrm dt}\begin{bmatrix}i_9\\v_3\\v_4\end{bmatrix}=\begin{bmatrix}0&-\frac{1}{L_9}&0\\\frac{1}{C_3}&0&0\\0&0&0\end{bmatrix}\begin{bmatrix}i_9\\v_3\\v_4\end{bmatrix}+\begin{bmatrix}0&0&0&\frac{1}{L_9}\\\frac{1}{C_3}&\frac{1}{C_3}&0&0\\-\frac{1}{C_4}&0&\frac{1}{C_4}&0\end{bmatrix}\begin{bmatrix}i_6\\i_7\\i_8\\v_5\end{bmatrix}$$

$$M=\begin{bmatrix}0&L^{-1}K_{11}\\-C^{-1}K_{11}^{\mathrm T}&0\end{bmatrix}=\left[\begin{array}{c:cc}0&-\frac{1}{L_9}&0\\ \hdashline\frac{1}{C_3}&0&0\\0&0&0\end{array}\right],$$

$$N=0,$$

$$
\boldsymbol{P}=\begin{bmatrix}0 & \boldsymbol{L}^{-1}\boldsymbol{K}_{13}\\ -\boldsymbol{C}^{-1}\boldsymbol{K}_{31}^{\mathrm{T}} & 0\end{bmatrix}=\left[\begin{array}{ccc:c}0 & 0 & 0 & \frac{1}{L_9}\\ \hdashline \frac{1}{C_3} & \frac{1}{C_3} & 0 & 0\\ -\frac{1}{C_4} & 0 & \frac{1}{C_4} & 0\end{array}\right],
$$

所以

$$
\frac{\mathrm{d}i_9}{\mathrm{d}t}=\frac{v_5-i_9}{L_9},\quad \frac{\mathrm{d}v_3}{\mathrm{d}t}=\frac{i_9+i_6+i_7}{C_3},\quad \frac{\mathrm{d}v_4}{\mathrm{d}t}=\frac{i_8-i_6}{C_4}.
$$

后面利用公式(1-20)，

$$
-\boldsymbol{R}^{-1}\boldsymbol{K}_{33}=\begin{bmatrix}-\frac{1}{R_6} & & 0\\ & -\frac{1}{R_7} & \\ 0 & & -\frac{1}{R_8}\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}=\begin{bmatrix}-\frac{1}{R_6}\\0\\0\end{bmatrix},
$$

$$
\boldsymbol{R}^{-1}\boldsymbol{K}_{31}=\begin{bmatrix}\frac{1}{R_6} & & 0\\ & \frac{1}{R_7} & \\ 0 & & \frac{1}{R_8}\end{bmatrix}\begin{bmatrix}-1 & 1\\ -1 & 0\\ 0 & -1\end{bmatrix}=\begin{bmatrix}-\frac{1}{R_6} & \frac{1}{R_6}\\ -\frac{1}{R_7} & 0\\ 0 & -\frac{1}{R_8}\end{bmatrix},
$$

$$
\boldsymbol{R}^{-1}\boldsymbol{K}_{32}=\begin{bmatrix}0 & 0\\ \frac{1}{R_7} & 0\\ 0 & \frac{1}{R_8}\end{bmatrix},
$$

$$
\boldsymbol{G}^{-1}\boldsymbol{K}_{33}^{\mathrm{T}}=R_5[1\quad 0\quad 0]=[R_5\quad 0\quad 0],
$$

$$
-\boldsymbol{G}^{-1}\boldsymbol{K}_{13}^{\mathrm{T}}=-R_5,\quad \boldsymbol{L}^{-1}\boldsymbol{K}_{11}=\frac{1}{L_9}[-1\quad 0]=[-\frac{1}{L_9}\quad 0]
$$

$$
-\boldsymbol{C}^{-1}\boldsymbol{K}_{11}^{\mathrm{T}}=\begin{bmatrix}-\frac{1}{C_3} & 0\\ 0 & -\frac{1}{C_4}\end{bmatrix}[-1\quad 0]^{\mathrm{T}}=\begin{bmatrix}-\frac{1}{C_3} & 0\\ 0 & -\frac{1}{C_4}\end{bmatrix}\begin{bmatrix}-1\\0\end{bmatrix}=\begin{bmatrix}\frac{1}{C_3}\\0\end{bmatrix},
$$

$$
-\boldsymbol{C}^{-1}\boldsymbol{K}_{31}^{\mathrm{T}}=\begin{bmatrix}-\frac{1}{C_3} & 0\\ 0 & -\frac{1}{C_4}\end{bmatrix}\begin{bmatrix}-1 & -1 & 0\\ 1 & 0 & -1\end{bmatrix}=\begin{bmatrix}\frac{1}{C_3} & \frac{1}{C_3} & 0\\ -\frac{1}{C_4} & 0 & \frac{1}{C_4}\end{bmatrix},
$$

$$\begin{bmatrix} I_{11} & -R^{-1}K_{33} \\ G^{-1}K_{33}^{\mathrm{T}} & I_{22} \end{bmatrix} = \left[\begin{array}{ccc:c} 1 & 0 & 0 & -\dfrac{1}{R_6} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdashline R_5 & 0 & 0 & 1 \end{array}\right],$$

$$\begin{bmatrix} 1 & 0 & 0 & -\dfrac{1}{R_6} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ R_5 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \dfrac{R_6}{R_5+R_6} & 0 & 0 & \dfrac{1}{R_5+R_6} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\dfrac{R_5R_6}{R_5+R_6} & 0 & 0 & \dfrac{R_6}{R_5+R_6} \end{bmatrix},$$

$$D = \begin{bmatrix} I_{11} & -R^{-1}K_{33} \\ G^{-1}K_{33} & I_{22} \end{bmatrix}^{-1} \begin{bmatrix} 0 & R^{-1}K_{31} \\ -G^{-1}K_{13}^{\mathrm{T}} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{R_6}{R_5+R_6} & 0 & 0 & \dfrac{1}{R_5+R_6} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \dfrac{-R_5R_6}{R_5+R_6} & 0 & 0 & \dfrac{R_6}{R_5+R_6} \end{bmatrix} \begin{bmatrix} 0 & -\dfrac{1}{R_6} & \dfrac{1}{R_6} \\ 0 & -\dfrac{1}{R_7} & 0 \\ 0 & 0 & -\dfrac{1}{R_8} \\ -R_5 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{-R_5}{R_5+R_6} & \dfrac{-1}{R_5+R_6} & \dfrac{1}{R_5+R_6} \\ 0 & -\dfrac{1}{R_7} & 0 \\ 0 & 0 & -\dfrac{1}{R_8} \\ \dfrac{-R_5R_6}{R_5+R_6} & \dfrac{R_5}{R_5+R_6} & \dfrac{-R_5}{R_5+R_6} \end{bmatrix}$$

$$E = \begin{bmatrix} I_{11} & -R^{-1}K_{33} \\ G^{-1}K_{33}^{\mathrm{T}} & I_{22} \end{bmatrix}^{-1} \begin{bmatrix} 0 & R^{-1}K_{23} \\ -C^{-1}K_{23} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{R_6}{R_5+R_6} & 0 & 0 & \dfrac{1}{R_5+R_6} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \dfrac{-R_5R_6}{R_5+R_6} & 0 & 0 & \dfrac{R_6}{R_5+R_6} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{1}{R_7} & 0 \\ 0 & 0 & \dfrac{1}{R_8} \\ 0 & 0 & 0 \end{bmatrix}$$

$$
=\begin{bmatrix}0&0&0\\0&\frac{1}{R_7}&0\\0&0&\frac{1}{R_8}\\0&0&0\end{bmatrix},
$$

$$
A=\begin{bmatrix}0&L^{-1}K_{11}\\-C^{-1}K_{11}^{T}&0\end{bmatrix}+\begin{bmatrix}0&L^{-1}K_{13}\\-C^{-1}K_{31}^{T}&0\end{bmatrix}D
$$

$$
=\begin{bmatrix}0&-\frac{1}{L_9}&0\\\frac{1}{C_3}&0&0\\0&0&0\end{bmatrix}+\begin{bmatrix}0&0&0&\frac{1}{L_9}\\\frac{1}{C_3}&\frac{1}{C_3}&0&0\\-\frac{1}{C_4}&0&\frac{1}{C_4}&0\end{bmatrix}\begin{bmatrix}\frac{-R_5}{R_5+R_6}&\frac{-1}{R_5+R_6}&\frac{1}{R_5+R_6}\\0&-\frac{1}{R_7}&0\\0&0&-\frac{1}{R_8}\\\frac{-R_5R_6}{R_5+R_6}&\frac{R_5}{R_5+R_6}&\frac{-R_5}{R_5+R_6}\end{bmatrix}
$$

$$
=\begin{bmatrix}0&-\frac{1}{L_9}&0\\\frac{1}{C_3}&0&0\\0&0&0\end{bmatrix}+\begin{bmatrix}\frac{-R_5R_6}{L_9(R_5+R_6)}&\frac{R_5}{L_9(R_5+R_6)}&\frac{-R_5}{L_9(R_5+R_6)}\\\frac{-R_5}{C_3(R_5+R_6)}&\frac{-R_5-R_6-R_7}{C_3R_7(R_5+R_6)}&\frac{1}{C_3(R_5+R_6)}\\\frac{R_5}{C_4(R_5+R_6)}&\frac{1}{C_4(R_5+R_6)}&\frac{-R_5-R_6-R_7}{C_4R_8(R_5+R_6)}\end{bmatrix}
$$

$$
=\begin{bmatrix}\frac{-R_5R_6}{L_9(R_5+R_6)}&\frac{-R_6}{L_9(R_5+R_6)}&\frac{-R_5}{L_9(R_5+R_6)}\\\frac{R_6}{C_3(R_5+R_6)}&\frac{-R_5-R_6-R_7}{C_3R_7(R_5+R_6)}&\frac{1}{C_3(R_5+R_6)}\\\frac{R_5}{C_4(R_5+R_6)}&\frac{1}{C_4(R_5+R_6)}&\frac{-R_5-R_6-R_7}{C_4R_8(R_5+R_6)}\end{bmatrix}.
$$

类似可计算 $\bar{B}$ 得

$$
\bar{B}=\begin{bmatrix}0&L^{-1}K_{12}\\-C^{-1}K_{21}^{T}&0\end{bmatrix}+\begin{bmatrix}0&L^{-1}K_{13}\\-C^{-1}K_{31}^{T}&0\end{bmatrix}E
$$

$$
=\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}+\begin{bmatrix}0&0&0&L_9\\\frac{1}{C_3}&\frac{1}{C_3}&0&0\\-\frac{1}{C_4}&0&\frac{1}{C_4}&0\end{bmatrix}\begin{bmatrix}0&0&0\\0&\frac{1}{R_7}&0\\0&0&\frac{1}{R_8}\\0&0&0\end{bmatrix}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{C_3 R_7} & 0 \\ 0 & 0 & \frac{1}{C_4 R_8} \end{bmatrix},
$$

$$
\frac{\mathrm{d}X}{\mathrm{d}t} = \bar{A}\ X + \bar{B}\ Y,
$$

其中

$$
Y = [0 \quad e_1(t) \quad e_2(t)]^{\mathrm{T}}.
$$

殊途同归,结果完全一致.

1.3　状态变量法的补充

前面讨论的状态变量法要求找到各边包含了所有电容及电压源的正常树,且仅包含电容及电压源.如若不然,比如电容过多或电感过多,都将超出讨论的范围.这时上述的办法还能适用吗?这一节将通过例子介绍如何将状态变量法予以扩充.

例 6　电路如图1-15所示.

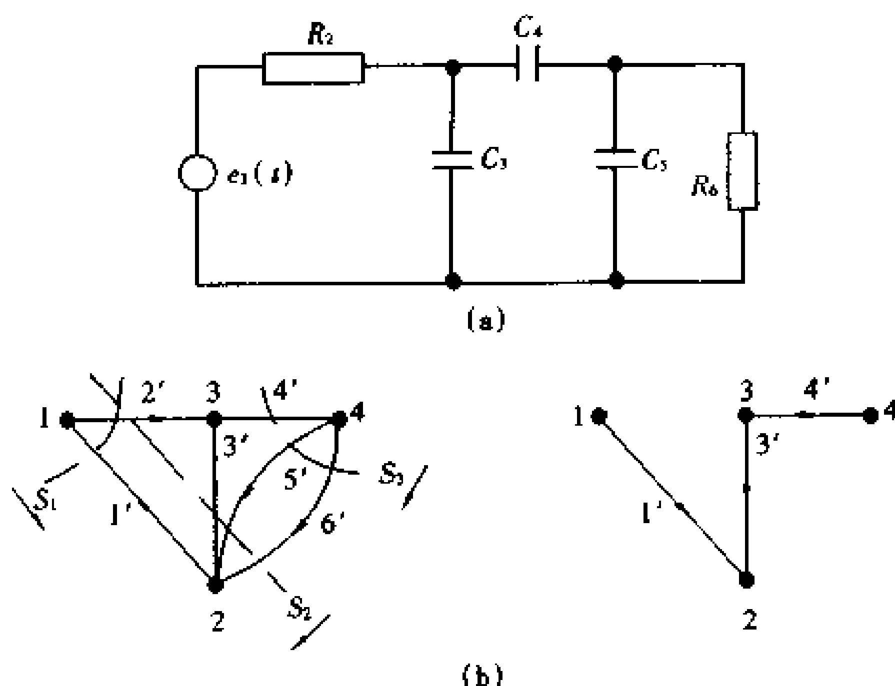


图 1-15

图 1-15 便是属于电容过多的例子,以至于不可能找到一正常树.

$$S_f = \begin{matrix} S1 \\ S2 \\ S3 \end{matrix}\left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 1 \\ 2' & 5' & 6' & 1' & 3' & 4' \end{array}\right],$$

$$K = S_1^{\mathrm{T}} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix}.$$

代入(1-9)式得

$$\begin{bmatrix} v_2 \\ v_5 \\ v_6 \\ i_1 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_2 \\ i_5 \\ i_6 \\ v_1 \\ v_3 \\ v_4 \end{bmatrix},$$

或

$$\begin{bmatrix} v_2 \\ v_5 \\ v_6 \\ i_1 \\ C_3 \dfrac{\mathrm{d}v_3}{\mathrm{d}t} \\ C_4 \dfrac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} G_2 v_2 \\ G_5 v_5 \\ C_6 \dfrac{\mathrm{d}v_6}{\mathrm{d}t} \\ e_1(t) \\ v_3 \\ v_4 \end{bmatrix}.$$

这时$\dfrac{\mathrm{d}v_6}{\mathrm{d}t}$出现在等号右端.由上得

$$\begin{cases} C_3 \dfrac{\mathrm{d}v_3}{\mathrm{d}t} = G_2 v_2 - G_5 v_5 - C_6 \dfrac{\mathrm{d}v_6}{\mathrm{d}t}, \\ C_4 \dfrac{\mathrm{d}v_4}{\mathrm{d}t} = G_5 v_5 + C_6 \dfrac{\mathrm{d}v_6}{\mathrm{d}t}, \end{cases} \tag{1-35}$$

但 $v_6 = v_3 - v_4$,于是有

$$\begin{aligned} \frac{\mathrm{d}v_6}{\mathrm{d}t} &= \frac{\mathrm{d}v_3}{\mathrm{d}t} - \frac{\mathrm{d}v_4}{\mathrm{d}t} \\ &= \frac{G_2}{C_3}v_2 - \frac{G_3}{C_3}v_3 - \frac{C_6}{C_3}\frac{\mathrm{d}v_6}{\mathrm{d}t} - \frac{G_5}{C_4}v_5 - \frac{C_6}{C_4}\frac{\mathrm{d}v_6}{\mathrm{d}t}, \end{aligned} \tag{1-36}$$

$$\left(1 + \frac{C_6}{C_3} + \frac{C_6}{C_4}\right)\frac{\mathrm{d}v_6}{\mathrm{d}t} = \frac{G_2}{C_3}v_2 - \frac{G_3}{C_3}v_3 - \frac{G_5}{C_4}v_5. \tag{1-37}$$

将(1-37)式代入(1-25)和(1-36)式从而消去右端的$\dfrac{\mathrm{d}v_6}{\mathrm{d}t}$.又 $v_2 = e_1(t) - v_3$,$v_5 = v_3 - v_4$,消去 v_2、v_5 后,最后得关于 v_3 和 v_4 的微分方程.

上述讨论用矩阵表示过程如下：

$$\begin{bmatrix} C_3 \frac{\mathrm{d}v_3}{\mathrm{d}t} \\ C_4 \frac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} G_2 & -G_5 \\ 0 & G_5 \end{bmatrix} \begin{bmatrix} v_2 \\ v_5 \end{bmatrix} + \begin{bmatrix} -C_6 \\ C_6 \end{bmatrix} \frac{\mathrm{d}v_6}{\mathrm{d}t}$$

或

$$\begin{bmatrix} C_3 & 0 \\ 0 & C_4 \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}v_3}{\mathrm{d}t} \\ \frac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} G_2 & 0 \\ 0 & G_5 \end{bmatrix} \begin{bmatrix} v_2 \\ v_5 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} C_6 \frac{\mathrm{d}v_6}{\mathrm{d}t}. \quad (1\text{-}38)$$

将

$$\begin{bmatrix} v_2 \\ v_5 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} e_1(t),$$

$$\frac{\mathrm{d}v_6}{\mathrm{d}t} = \frac{\mathrm{d}v_3}{\mathrm{d}t} - \frac{\mathrm{d}v_4}{\mathrm{d}t} = [1 \quad -1] \begin{bmatrix} \frac{\mathrm{d}v_3}{\mathrm{d}t} \\ \frac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix}$$

代入(1-38)式得，

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{C_3} & 0 \\ 0 & \frac{1}{C_4} \end{bmatrix} \left(\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} G_2 & 0 \\ 0 & G_5 \end{bmatrix} \left[\begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} + \right.\right.$$

$$\left.\left. \begin{bmatrix} 1 \\ 0 \end{bmatrix} e_1(t) \right] + \begin{bmatrix} -1 \\ 1 \end{bmatrix} [C_6 \quad -C_6] \begin{bmatrix} \frac{\mathrm{d}v_3}{\mathrm{d}t} \\ \frac{\mathrm{d}v_4}{\mathrm{d}t} \end{bmatrix} \right).$$

又

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} [C_6 - C_6] = \begin{bmatrix} -C_6 & C_6 \\ C_6 & -C_6 \end{bmatrix},$$

所以

$$\begin{bmatrix} \frac{1}{C_3} & 0 \\ 0 & \frac{1}{C_4} \end{bmatrix} \begin{bmatrix} -C_6 & C_6 \\ C_6 & -C_6 \end{bmatrix} = \begin{bmatrix} -\frac{C_6}{C_3} & \frac{C_6}{C_3} \\ \frac{C_6}{C_4} & -\frac{C_6}{C_4} \end{bmatrix},$$

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -\frac{C_6}{C_3} & \frac{C_6}{C_3} \\ \frac{C_6}{C_4} & -\frac{C_6}{C_4} \end{bmatrix} \right) \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{C_3} & -\frac{1}{C_3} \\ 0 & \frac{1}{C_4} \end{bmatrix} \begin{bmatrix} -G_2 & 0 \\ G_5 & -G_5 \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_3} & -\frac{1}{C_3} \\ 0 & \frac{1}{C_4} \end{bmatrix} \begin{bmatrix} G_2 \\ 0 \end{bmatrix} e_1(t)$$

或

$$\begin{bmatrix} 1-\frac{C_6}{C_3} & -\frac{C_6}{C_3} \\ -\frac{C_6}{C_4} & 1+\frac{C_6}{C_4} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{C_3 R_2} - \frac{1}{C_3 R_5} & -\frac{1}{C_3 R_5} \\ \frac{1}{C_4 R_5} & -\frac{1}{C_4 R_5} \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_3 R_2} \\ 0 \end{bmatrix} e_1(t),$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1-\frac{C_6}{C_3} & -\frac{C_6}{C_3} \\ -\frac{C_6}{C_4} & 1+\frac{C_6}{C_4} \end{bmatrix}^{-1} \begin{bmatrix} \frac{-1}{C_3 R_2} - \frac{1}{C_3 R_5} & -\frac{1}{C_3 R_5} \\ \frac{1}{C_4 R_5} & \frac{1}{C_4 R_5} \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} +$$

$$\begin{bmatrix} 1-\frac{C_6}{C_3} & -\frac{C_6}{C_3} \\ -\frac{C_6}{C_4} & 1+\frac{C_6}{C_4} \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{C_3 R_2} \\ 0 \end{bmatrix} e_1(t).$$

解出 v_3、v_4，可求得 $v_6 = v_5 = v_3 - v_4, v_2 = e_1(t) v_3, i_1 = \frac{v_2}{R_2}$.

例 7　解图1-16(a)所示电路.

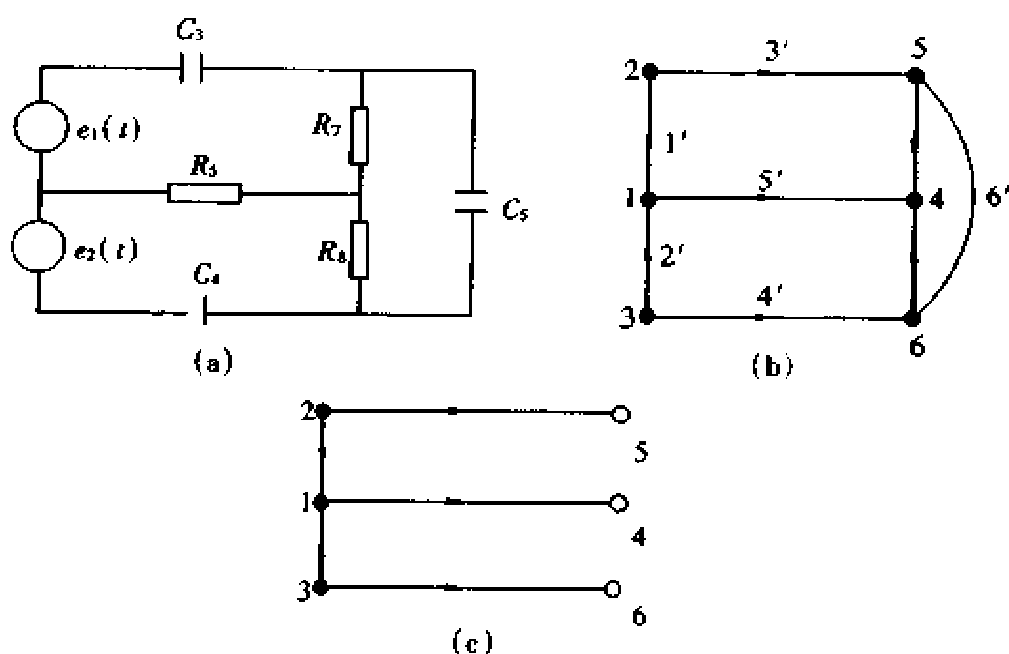


图 1-16

图 1-16(b) 的关联矩阵为

$$
\boldsymbol{B}=\begin{array}{c}1\\2\\3\\4\\5\\6\\ \\\end{array}\!\!\begin{array}{c}\left[\begin{array}{ccc:ccccc}0&0&0&-1&-1&0&0&1\\0&0&0&1&0&1&0&0\\0&0&0&0&1&0&1&0\\0&-1&-1&0&0&0&0&-1\\1&1&0&0&0&-1&0&0\\-1&0&1&0&0&0&-1&0\end{array}\right]\\ \begin{array}{ccc:ccccc}6'&7'&8'&1'&2'&3'&4'&5'\end{array}\end{array},
$$

于是有

$$
\boldsymbol{B}_{(1)}=\begin{array}{c}2\\3\\4\\5\\6\\ \\\end{array}\!\!\begin{array}{ccc:ccccc}0&0&0&1&0&1&0&0\\0&0&0&0&1&0&1&0\\0&-1&-1&0&0&0&0&-1\\1&1&0&0&0&-1&0&0\\-1&0&1&0&0&0&-1&0\\6'&7'&8'&1'&2'&3'&4'&5'\end{array},
$$

$$
\boldsymbol{B}_2=\begin{bmatrix}1&0&1&0&0\\0&1&0&1&0\\0&0&0&0&-1\\0&0&-1&0&0\\0&0&0&-1&0\end{bmatrix}.
$$

通过约当法求得

$$
\boldsymbol{B}_2^{-1}=\begin{bmatrix}1&0&0&1&0\\0&1&0&0&1\\0&0&0&-1&0\\0&0&0&0&-1\\0&0&-1&0&0\end{bmatrix},
$$

和

$$
\boldsymbol{K}=\boldsymbol{B}_1^{\mathrm{T}}(\boldsymbol{B}_2^{-1})^{\mathrm{T}}=\begin{bmatrix}0&0&0&1&-1\\0&0&-1&1&0\\0&0&-1&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&0&-1\\1&0&-1&0&0\\0&1&0&-1&0\end{bmatrix}
$$

$$
=\begin{bmatrix}1&-1&-1&1&0\\1&0&-1&0&1\\0&1&0&-1&1\end{bmatrix}.
$$

令

$$V_{\bar{T}} = \begin{bmatrix} v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} v_6 \\ R_7 i_7 \\ R_8 i_8 \end{bmatrix}, \qquad I_{\bar{T}} = \begin{bmatrix} C_8 \dfrac{dv_6}{dt} \\ i_7 \\ i_8 \end{bmatrix},$$

$$I_T = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ C_3 \dfrac{dv_3}{dt} \\ C_4 \dfrac{dv_4}{dt} \\ G_5 v_5 \end{bmatrix}, \quad V_T = \begin{bmatrix} e_1(t) \\ e_2(t) \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}$$

因为

$$\begin{bmatrix} V_{\bar{T}} \\ I_T \end{bmatrix} = \begin{bmatrix} 0 & K \\ -K^{T} & 0 \end{bmatrix} \begin{bmatrix} I_{\bar{T}} \\ V_T \end{bmatrix},$$

即

$$\begin{bmatrix} v_6 \\ R_7 i_7 \\ R_8 i_8 \\ i_1 \\ i_2 \\ C_3 \dfrac{dv_3}{dt} \\ C_4 \dfrac{dv_4}{dt} \\ G_5 v_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} C_6 \dfrac{dv_6}{dt} \\ i_7 \\ i_8 \\ e_1(t) \\ e_2(t) \\ v_3 \\ v_4 \\ v_5 \end{bmatrix},$$

$\dfrac{dv_6}{dt}$ 出现在等式右端.由上得

$$C_3 \frac{dv_3}{dt} = C_6 \frac{dv_6}{dt} + i_7, \tag{1-39}$$

$$C_4 \frac{dv_4}{dt} = -C_6 \frac{dv_6}{dt} + i_8, \tag{1-40}$$

但 $v_6 = e_1(t) - e_2(t) - v_3 + v_4$,于是有

$$\begin{aligned} \frac{dv_6}{dt} &= e'_1(t) - e'_2(t) - \frac{dv_3}{dt} + \frac{dv_4}{dt} \\ &= e'_1(t) - e'_2(t) - \left[\frac{1}{C_3} C_6 + \frac{C_6}{C_4}\right] \frac{dv_6}{dt} + \frac{1}{C_3} i_7 + \frac{1}{C_4} i_8, \end{aligned}$$

$$\left(1 + \frac{C_6}{C_3} + \frac{C_6}{C_4}\right) \frac{dv_6}{dt} = e'_1(t) - e'_2(t) + \frac{1}{C_3} i_7 + \frac{1}{C_4} i_8.$$

又

$$i_7 = \frac{e_1(t) - v_3 + v_5}{R_7}, \tag{1-41}$$

$$i_8 = \frac{e_2(t) - v_4 + v_5}{R_8}, \tag{1-42}$$

$$v_5 = -R_5(i_7 + i_8), \tag{1-43}$$

消去 v_5 得

$$\begin{cases} i_7 = \dfrac{e_1(t) - v_3 - R_5 i_7 - R_5 i_8}{R_7}, \\ i_8 = \dfrac{e_2(t) - v_4 - R_5 i_7 - R_5 i_8}{R_8}, \end{cases}$$

或

$$\begin{cases} \left(1 + \dfrac{R_5}{R_7}\right) i_7 + \dfrac{R_5}{R_7} i_8 = \dfrac{1}{R_7} e_1(t) - \dfrac{1}{R_7} v_3 \\ \dfrac{R_5}{R_8} i_7 + \left(1 + \dfrac{R_5}{R_8}\right) i_8 = \dfrac{1}{R_8} e_2(t) - \dfrac{1}{R_8} v_4 \end{cases}$$

解方程组得

$$i_7 = \begin{vmatrix} 1/R_7[e_1(t) - v_3] & R_5/R_7 \\ 1/R_8[e_2(t) - v_4] & 1 + R_5/R_8 \end{vmatrix} \Bigg/ \begin{vmatrix} 1 + R_5/R_7 & R_5/R_7 \\ R_5/R_8 & 1 + R_5/R_8 \end{vmatrix}$$

$$= \frac{-(R_8 + R_5)v_3 + R_5 v_4 + (R_8 + R_5)e_1(t) - R_5 e_2(t)}{R_7 R_8 + R_5 R_7 + R_5 R_8}, \tag{1-44}$$

$$i_8 = \frac{R_5 v_3 - (R_7 + R_5)v_4 + (R_7 + R_5)e_1(t) - R_5 e_1(t)}{R_7 R_8 + R_5 R_7 + R_8 R_5}. \tag{1-45}$$

将 i_7、i_8 代入(1-39)、(1-40) 式,分别得

$$(C_3 + C_6)\frac{dv_3}{dt} - C_6 \frac{dv_4}{dt}$$
$$= \frac{-(R_5 + R_8)(v_3 - e_1(t)) + R_5(v_4 - e_2(t))}{R_7 R_8 + R_5 R_8 + R_5 R_7} + C_6(e'_1(t) - e'_2(t)). \tag{1-46}$$

和

$$-C_6 \frac{dv_3}{dt} + (C_4 + C_6)\frac{dv_4}{df}$$
$$= \frac{R_5(v_3 - e(t)) - (R_7 + R_5)(v_4 - e_2(t))}{R_7 R_8 + R_5 R_7 + R_5 R_8} + C_6(e'_2(t) - e'_1(t)) \tag{1-47}$$

把(1-46) 和(1-47) 式写成矩阵形式得

$$\begin{bmatrix} C_3 + C_6 & -C_6 \\ -C_6 & C_4 + C_6 \end{bmatrix} \begin{bmatrix} \dfrac{dv_3}{dt} \\ \dfrac{dv_4}{dt} \end{bmatrix}$$

$$= \frac{1}{R_5R_7+R_5R_8+R_7R_8}\begin{bmatrix} -R_5-R_8 & R_5 \\ R_5 & -R_5-R_7 \end{bmatrix}\left(\begin{bmatrix} v_3 \\ v_4 \end{bmatrix} - \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}\right)$$

$$+ C_6\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} e'_1(t) \\ e'_2(t) \end{bmatrix}$$

或写成

$$\frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} = \boldsymbol{AX} + \boldsymbol{U}, \tag{1-48}$$

其中

$$\boldsymbol{X} = (v_3, v_4)^{\mathrm{T}},$$

$$\boldsymbol{A} = \frac{1}{R_5R_7+R_5R_8+R_7R_8}\begin{bmatrix} C_3+C_6 & -C_6 \\ -C_6 & C_4+C_6 \end{bmatrix}^{-1}\begin{bmatrix} -R_5-R_8 & R_5 \\ R_5 & -R_5-R_7 \end{bmatrix},$$

$$\boldsymbol{U} = C_6\begin{bmatrix} C_3+C_6 & -C_6 \\ -C_6 & C_4+C_6 \end{bmatrix}^{-1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} e'_1(t) \\ e'_2(t) \end{bmatrix} - A\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}. \tag{1-49}$$

解(1-48)式得 v_3、v_4,可从(1-44)、(1-45)式求 i_7、i_8;从(1-43)式求得 v_5,再求出 $v_6 = e_1(t) - e_2(t) - v_3 + v_4$,进一步解得 i_1、i_2.

2　信号流图

前面已讨论到的许多电路问题,它的各边的电流、电压等的拉普拉斯变换满足某一代数方程组.这一章介绍的信号流图就是将代数方程组问题转化为一种带权的有向图,并讨论其图的解法.特别是电路问题等物理系统可以根据其物理特性直接构造其信号流图,进而求出解.所以信号流图是一种对研究线性系统十分直观、使用很方便的方法.

2.1　玛逊信号流图

将代数方程组转化为带权的有向图玛逊(Mason)信号流图的办法如下

设代数方程组为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3. \end{cases} \tag{2-1}$$

将方程组(2-1)改写为

$$\begin{cases} x_1 = b_1 + b_{11}x_1 + b_{12}x_2 + b_{13}x_3, \\ x_2 = b_2 + b_{21}x_1 + b_{22}x_2 + b_{23}x_3, \\ x_3 = b_3 + b_{31}x_1 + b_{32}x_2 + b_{33}x_3. \end{cases} \tag{2-2}$$

图的表示规则:

用 $x_i \xrightarrow{a} x_j$ 表示 $x_j = ax_i$

用 $x_i \xrightarrow{a} x_k$，$x_j \xrightarrow{b} x_k$ 表示 $x_k = ax_i + bx_j$.

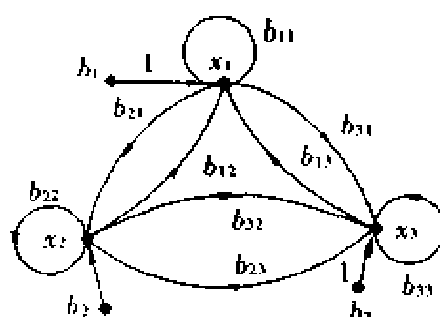


图 2-1

于是方程组(2-2)，实际上即方程组(2-1) 可表为图 2-1.

例 1 化下述方程组为带权有向图：

$$\begin{cases} x_1 - x_2 + x_3 = 0, \\ 2x_1 + 2x_2 + x_3 = 0, \\ -x_1 - x_2 + x_3 = 0. \end{cases}$$

先转化方程组为

$$\begin{cases} x_1 = x_2 - x_3, \\ x_2 = 2x_1 + 3x_2 + x_3, \\ x_3 = x_1 + x_2. \end{cases}$$

作有向图如图 2-2 所示.

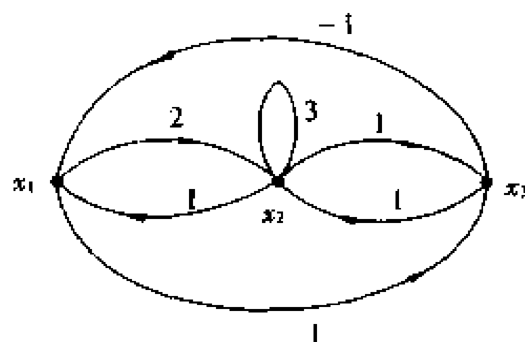


图 2-2

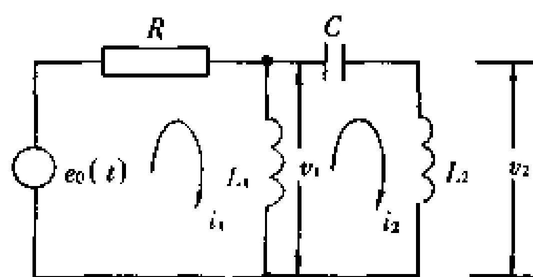


图 2-3

例 2 某电路如图2-3所示. 根据基尔霍夫定律有

$$e_0 = Ri_1 + v_1, \quad v_1 = \frac{1}{C}\int i_2 \mathrm{d}t + v_2,$$

$$v_1 = L_1 \frac{\mathrm{d}}{\mathrm{d}t}(i_1 - i_2), \quad v_2 = L_2 \frac{\mathrm{d}i_2}{\mathrm{d}t}.$$

设 $\mathscr{L}\{e_0(t)\} = E_0(p)$，$\mathscr{L}\{i_k\} = I_k(p), k = 1,2, \mathscr{L}\{v_h\} = V_h(p)$，$h = 1,2.$ 对上面的等式作拉普拉斯变换得

$$E_0(p) = RI_1(p) + V_1(p),$$

$$V_1(p) = \frac{1}{Cp}I_2(p) + V_2(p),$$

$$V_1(p)L_1p[I_1(p) - I_2(p)], \quad V_2(p) = L_2pI_2(p),$$

改写为

$$
(\text{I})\quad \begin{cases} I_1(p) = \dfrac{1}{R}[E_0(p) - V_1(p)], \\ I_2(p) = Cp[V_1(p) - V_2(p)], \\ V_1(p) = L_1p[I_1(p) - I_2(p)], \\ V_2(p) = L_2pI_2(p). \end{cases}
$$

还可将上述关系重新排列得

$$
(\text{II})\quad \begin{cases} I_1 = \dfrac{1}{R}[E_0 - V_1], \\ I_2 = Cp[V_1 - V_2], \\ V_1 = L_1p[I_1 - I_2], \\ V_2 = L_2pI_2. \end{cases}
$$

根据(Ⅱ)可作信号流图如图 2-4.

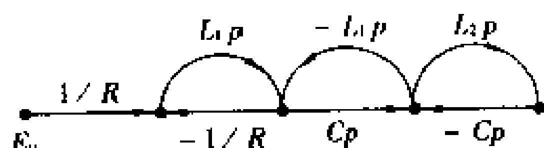


图 2-4

例 3 电路如图2-5所示，根据基尔霍夫定律有

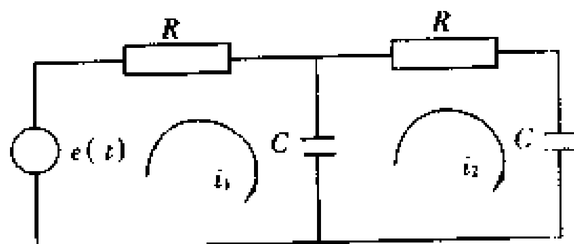


图 2-5

$$
Ri_1 + \frac{1}{C}\int_0^t (i_1 - i_2)\mathrm{d}t = e(t),
$$

$$
Ri_2 + \frac{1}{C}\int_0^t i_2\mathrm{d}t + \frac{1}{C}\int_0^t (i_2 - i_1)\mathrm{d}t = 0,
$$

$$
v_2 = \frac{1}{C}\int_0^t i_2\mathrm{d}t.
$$

对上式作拉普拉斯变换得

$$
RI_2 + \frac{1}{Cp}(I_1 - I_2) = E(p),
$$

$$
RI_2 + \frac{1}{Cp}I_2 + \frac{1}{Cp}(I_2 - I_1) = 0.
$$

$$V_2 = \frac{1}{Cp}I_2,$$

或

$$I_1 = \frac{1}{R+\frac{1}{Cp}}E + \frac{\frac{1}{Cp}}{R+\frac{1}{Cp}}I_2 = \frac{Cp}{RCp+1}E + \frac{1}{RCp+1}I_2$$

$$I_2 = \frac{1}{Cp(R+\frac{2}{Cp})}I_1 = \frac{1}{RCp+2}I_1, \quad V_2 = \frac{1}{Cp}I_2.$$

可得信号流图如图 2-6.

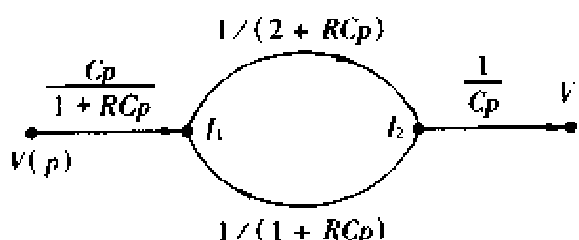


图 2-6

2.2 信号流图的运算规则

1. 加法规则

两个顶点间 m 条边同向,可代以一条与它们方向相同的边,该边的权为 m 条同向边的权的和.如图 2-7.

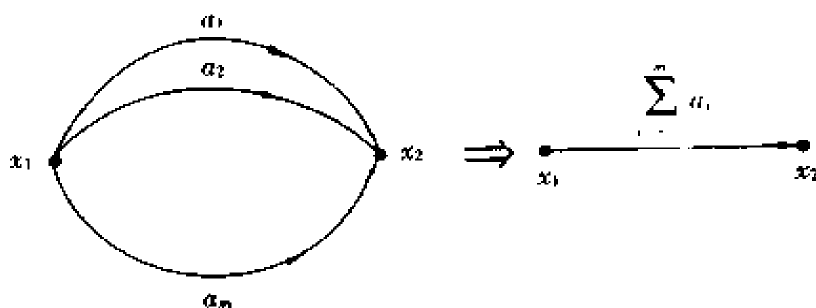


图 2-7

2. 乘法规则

$m+1$ 个顶点 $x_1, x_2, \cdots, x_{m+1}$,若有有向边$(x_1, x_2), (x_2, x_3), \cdots, (x_m, x_{m+1})$,而且设$(x_i, x_{i+1})$边的权为 $a_i, i = 1, 2, \cdots, m$,则可用一有向边(x_1, x_{m+1})代替它,而且(x_1, x_{m+1})的权为 $a_1 \cdot a_2 \cdots a_m$,即为$(x_1, x_2), (x_2, x_3), \cdots, (x_m, x_{m+1})$边的权的乘积.如图 2-8.

3. 顶点消去法

先举一例子,设

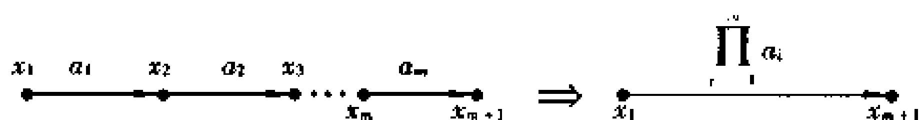


图 2-8

$$x_3 = a_1x_1 + a_2x_2,$$
$$y_1 = b_1x_3 = b_1a_1x_1 + b_1a_2x_2,$$
$$y_2 = b_2x_3 = b_2a_1x_1 + b_2a_2x_2.$$

消去 x_3 点的过程如图 2-9：

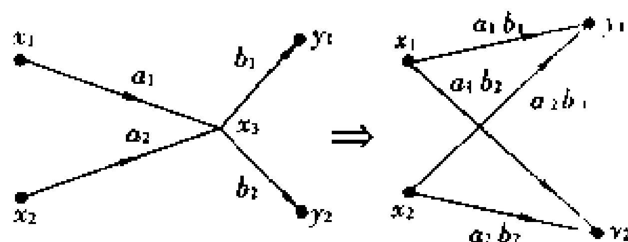


图 2-9

从而消去 x_3 点.

上述结果可以推及一般，即若存在以 x 点为终点的带权 a_i 的边 (x_i, x)，$i = 1, 2, \cdots, m$，以及以 x 为始点的带权 b_j 的边 (x, y_j)，$j = 1, 2, \cdots, n$，则可消 x 点得带权 a_ib_j 的边 (x_i, y_j)，$i = 1, 2, \cdots, m$，$j = 1, 2, \cdots, n$.

4. 自环消去法

先从一个实例入手，可得一般规律. 已知

$$x_2 = a_1x_1 + a_2x_2, \quad x_3 = a_3x_2, \quad x_4 = a_4x_2,$$

但 $(1 - a_2)x_2 = a_1x_1$，所以

$$x_2 = \frac{a_1x_1}{1 - a_2}, \quad x_3 = \frac{a_3a_1x_1}{1 - a_2}, \quad x_4 = \frac{a_4a_1x_1}{1 - a_2}.$$

将这个推演过程可用图的形式表达如下(图 2-10)：

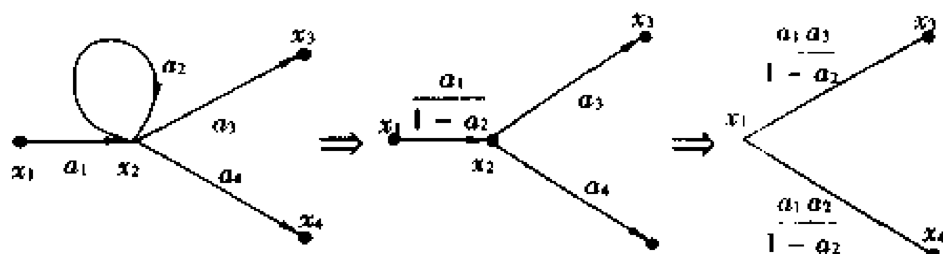


图 2-10

又如

$$x_3 = ax_1 + bx_2 + dx_3, \quad x_4 = cx_3.$$

所以$(1-d)x_3 = ax_1 + bx_2$，

$$x_4 = \frac{ca}{1-d}x_1 + \frac{cb}{1-d}x_2.$$

推广过程可用图 2-11 表示如下：

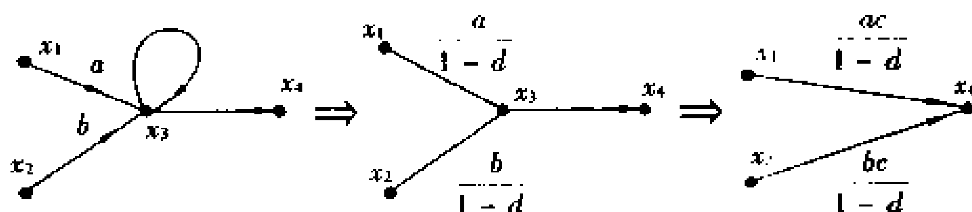


图 2-11

一般有如图 2-12 的结果.

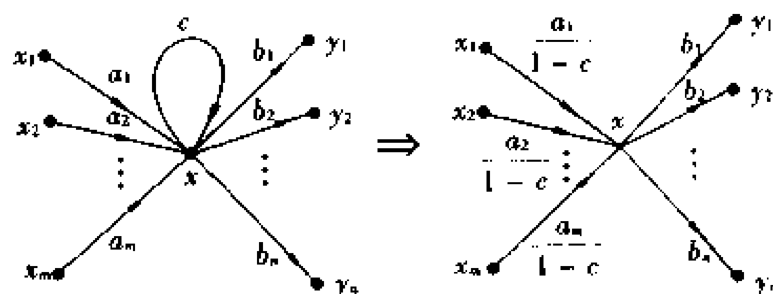


图 2-12

最后消去 x 点得常数$\frac{a_i b_j}{1-c}$的边(x_i, y_j)，$i = 1,2,\cdots,m$，$j = 1,2,\cdots,n$；构成常数的二分图.

5. 逆向

若 $x_1 = ax_2 + bx_3$，则

$$x_2 = \frac{1}{a}x_1 - \frac{b}{a}x_3.$$

用图来表示为(图 2-13)

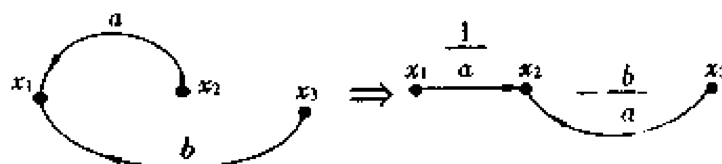


图 2-13

即权为 a 的有向边(x_2, x_1)可改为有向边(x_1, x_2)，但权为$\frac{1}{a}$，权为 b 的有向边(x_3, x_1)改为权为 $-\frac{b}{a}$ 的有向边(x_3, x_2).

同样可推广为一般，即如图 2-14 所示.

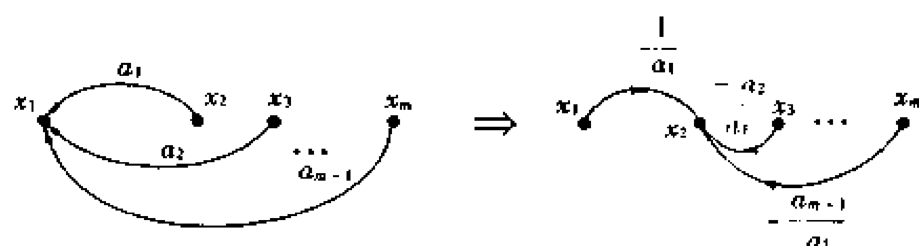


图 2-14

现在罗列常见的图的算法规则如下：

(1) $x_1 \xrightarrow{a_1} x_2$，$x_1 \xrightarrow{a_2} x_2$ $\Rightarrow$ $x_1 \xrightarrow{a_1+a_2} x_2$

(2) $x_1 \xrightarrow{a_1} x_2 \xrightarrow{a_2} x_3$ $\Rightarrow$ $x_1 \xrightarrow{a_1 a_2} x_3$

(3) $x_1 \xrightarrow{a_1} x_3$，$x_2 \xrightarrow{a_2} x_3$，$x_3 \xrightarrow{a_3} x_4$ $\Rightarrow$ $x_1 \xrightarrow{a_1 a_3} x_4$，$x_2 \xrightarrow{a_2 a_3} x_4$

(4) $x_1 \xrightarrow{a_1} x_2 \xrightarrow{a_2} x_3$，$x_3 \xrightarrow{a_3} x_2$ $\Rightarrow$ $x_1 \xrightarrow{a_1 a_2} x_3$，$x_3$ 自环 $a_2 a_3$

(5) $x_1 \xrightarrow{a_1} x_2$，$x_2$ 自环 a_2，$x_2 \xrightarrow{a_3} x_3$，$x_2 \xrightarrow{a_4} x_4$ $\Rightarrow$ $x_1 \xrightarrow{\frac{a_1}{1-a_2}} x_2$，$x_2 \xrightarrow{a_3} x_3$，$x_2 \xrightarrow{a_4} x_4$ $\Rightarrow$ $x_1 \xrightarrow{\frac{a_1 a_3}{1-a_2}} x_3$，$x_1 \xrightarrow{\frac{a_1 a_3}{1-a_4}} x_4$

(6) $x_1 \xrightarrow{a_1} x_2$，$x_2$ 自环 a_2，$x_2 \xrightarrow{a_3} x_3$ $\Rightarrow$ $x_1 \xrightarrow{\frac{a_1}{1-a_2}} x_2 \xrightarrow{a_3} x_3$ $\Rightarrow$ $x_1 \xrightarrow{\frac{a_1 a_3}{1-a_2}} x_3$

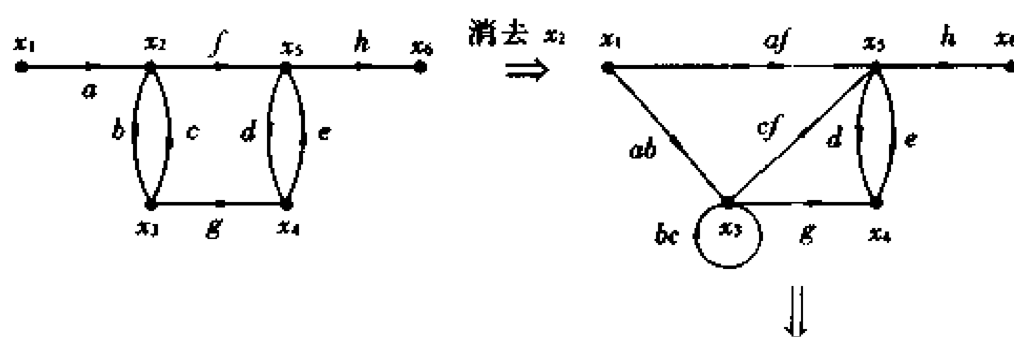
(7)

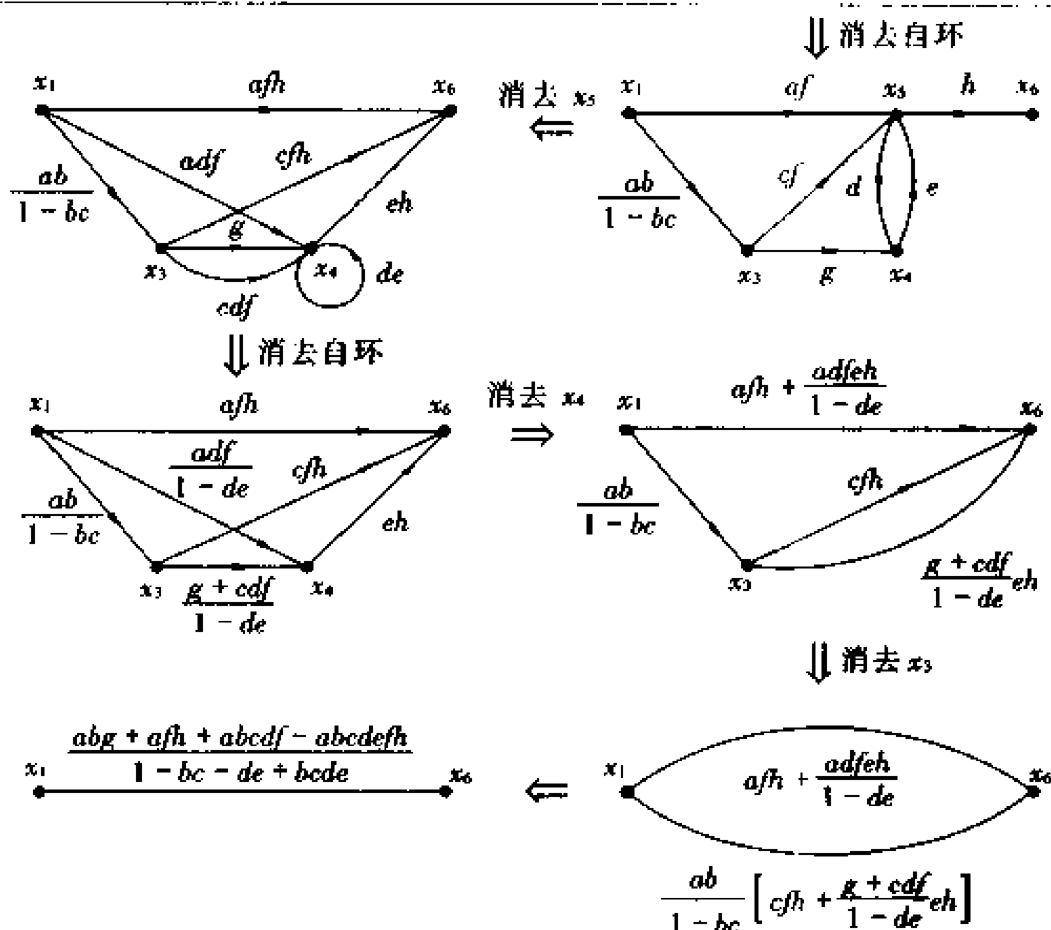
(8)

(9)

(10)

例 4





$$\begin{cases} x_2 = ax_1 + cx_3, & (1a) \\ x_3 = bx_2, & (2a) \\ x_4 = gx_3 + dx_5, & (3a) \\ x_5 = fx_2 + ex_4, & (4a) \\ x_6 = hx_5, & (5a) \end{cases} \xrightarrow{\text{消去}x_2} \begin{cases} x_3 = bax_1 + bcx_3, & (1b) \\ x_4 = gx_3 + dx_5, & (2b) \\ x_5 = afx_1 + fcx_3 + ex_4, & (3b) \\ x_6 = hx_5. & (4b) \end{cases}$$

从(1b)式得

$$(1-bc)x_3 = abx_1,$$

代入得

$$\begin{cases} x_3 = \dfrac{ab}{1-bc}x_1, & (1c) \\ x_4 = gx_3 + dx_5, & (2c) \\ x_5 = afx_1 + cfx_3 + ex_4, & (3c) \\ x_6 = hx_5, & (4c) \end{cases}$$

得

$$x_6 = hx_5 = ahfx_1 + cfhx_3 + ehx_4,$$

$$x_4 = gx_3 + dx_5 = gx_3 + adfx_1 + cdfx_3 + edx_4,$$

$$(1-ed)x_4 = gx_3 + adfx_1 + cdfx_3,$$

$$x_4 = \frac{g+cdf}{1-de}x_3 + \frac{adf}{1-de}x_1 = \frac{g+cdf}{1-de}x_3 + \frac{adf}{1-de}x_1.$$

再由

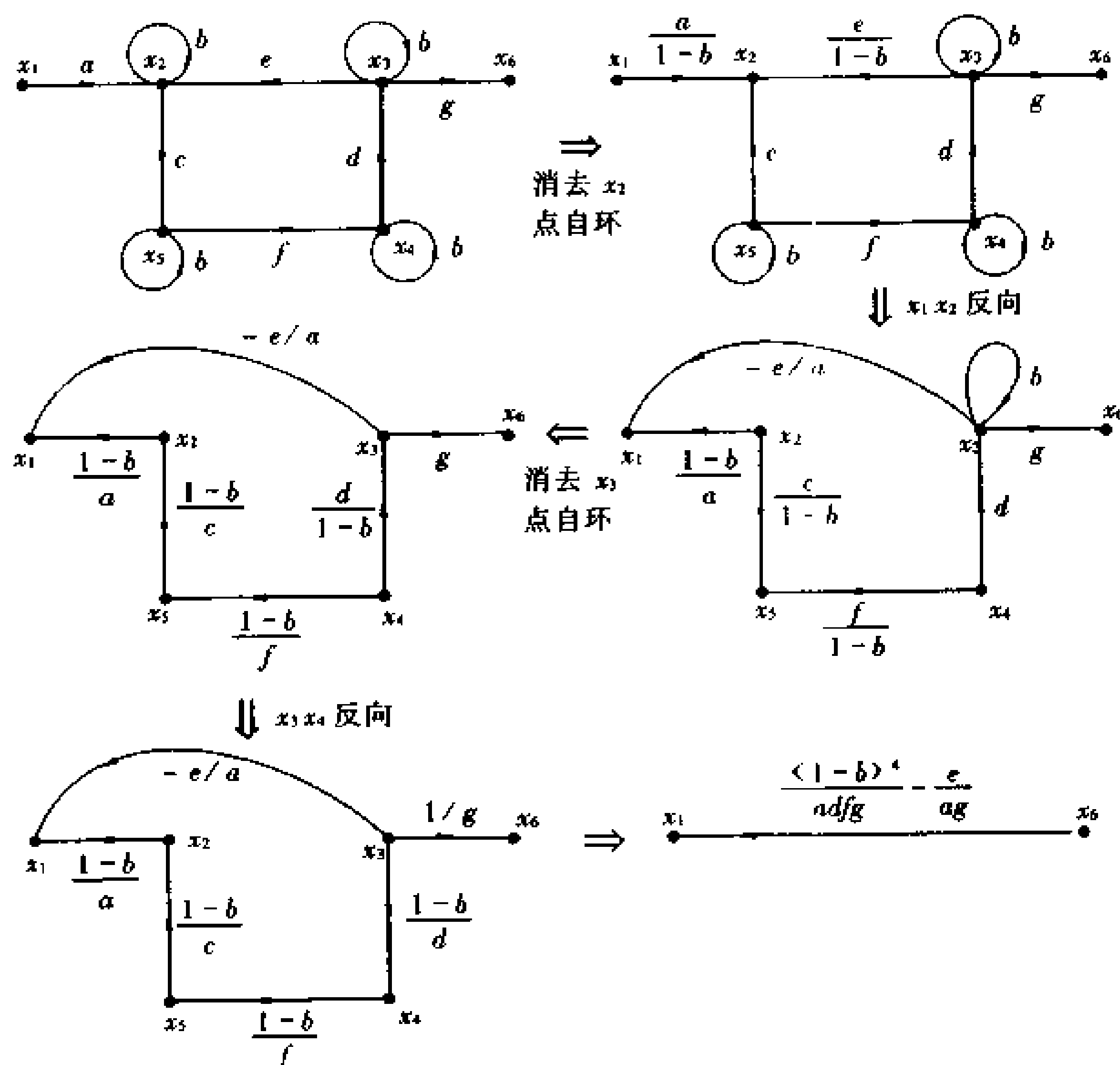
$$\begin{cases} x_3 = \dfrac{ab}{1-be}x_1, & (1e) \\ x_4 = \dfrac{adf}{1-de}x_1 + \left(\dfrac{g+cdf}{1-de}\right)x_3, & (2e) \\ x_6 = afhx_1 + cfhx_3 + ehx_4, & (3e) \end{cases}$$

将(2e)式代入(3e)式得

$$\begin{cases} x_3 = \dfrac{ab}{1-be}x_1, & (1g) \\ x_6 = \left(afh + \dfrac{adfeh}{1-de}\right)x_1 + \left(\dfrac{g+cdf}{1-de}eh + cfh\right)x_3. & (2g) \end{cases}$$

(1e)、(2e)等式为第5步图对应的方程.其解不另叙述.通过以上过程可以理解图算法的实质.

例5



2.3　科特斯方法

2.3.1　科特斯信号流图

假定现在讨论的方程组为

$$
\begin{cases}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 x_{n+1}, \\
a_{21}x_2 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 x_{n+1}, \\
\cdots\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n x_{n+1},
\end{cases}
\tag{2-3}
$$

或记为

$$AX = bx_{n+1}.$$

令 A^* 表示在矩阵 A 的右端加上一列 $-b$,在它的下方补上一行 0 的$(n+1)\times(n+1)$ 加边方阵.构造**科特斯信号流图** $G_C(A)$ 如下:

$G_C(A^*)$ 图有 $n+1$ 个顶点:

$$x_1, x_2, \cdots, x_n, x_{n+1}.$$

当 $a_{ji} \neq 0$时,从 x_i 到x_j引一条有向边(x_i, x_j),权为 $a_{ji}, i, j = 1,2,\cdots,n,n+1$.显然$A^*$ 是图 $G_C(A^*)$的带权邻接矩阵的转置.$G_C(A^*)$称为关于 A^* 的科特斯信号流图,或(2-3) 式的科特斯信号流图.

例如方程

$$
\begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & -2 \\ 2 & 1 & 2 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} x_4,
$$

其

$$
A^* = \begin{bmatrix} 2 & 1 & 1 & -1 \\ 1 & -1 & 2 & 0 \\ 2 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},
$$

则 $G_C(A^*)$ 如图 2-15 所示:

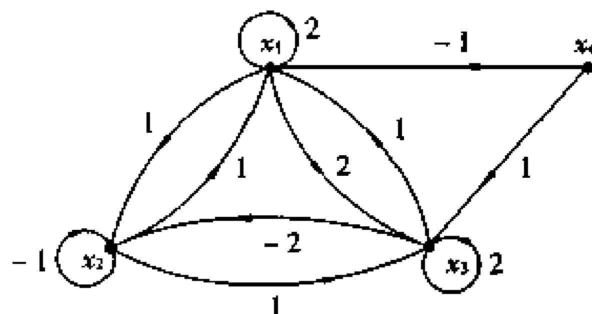


图 2-15

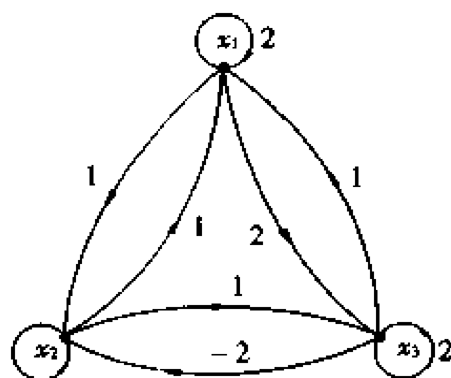


图 2-16

对于 $G_C(A^*)$ 图的 x_i 点，$1 \leqslant i \leqslant n$，进入 x_i 的每一条边的权与出发点变量之积的和等于零. 它是方程组的第 i 个方程. 从 $G_C(A^*)$ 图中消去 x_{n+1} 点便得矩阵 A 的科特斯信号流图(图 2-16).

2.3.2　1- 因子子图

$G = (V, E)$ 是一有向图，$G^{(1)} = (V, E^{(1)})$ 是 G 的支撑子图，且 $G^{(1)}$ 的所有顶点入度和出度都为 1. 则称 $G^{(1)}$ 图是 G 的 **1- 因子子图**.

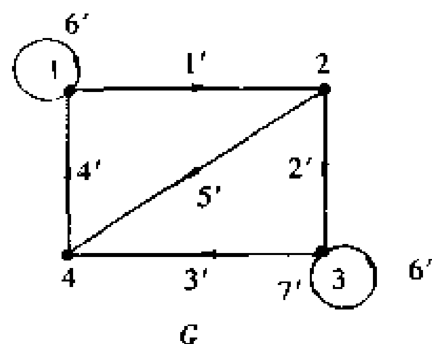


图 2-17

$G^{(1)}$ 图包含了 G 图的所有顶点，且包含了 G 图所有不相交的回路及自环.

例如图 G 如图 2-17 所示，则图 2-18 中 $G_1^{(1)}$ 和 $G_2^{(1)}$ 是 G 的两个 1- 因子子图.

从邻接矩 $A = [a_{ij}]_{n\times n}$ 的定义可知，对于矩阵

$$A^r = [a_{ij}^r]_{n\times n} = \underbrace{A \times A \times \cdots \times A}_{r\text{次}},$$

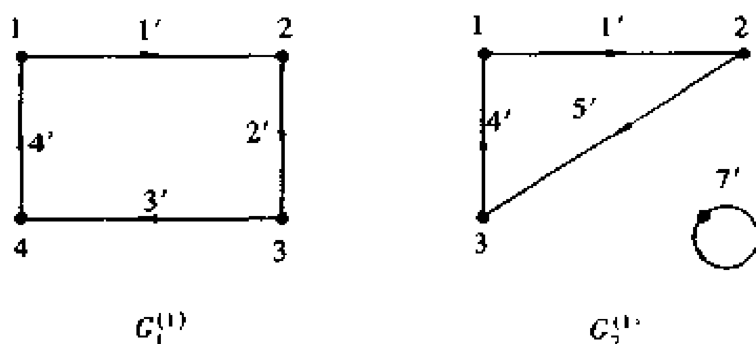


图 2-18

有

$$a_{ij}^{(r)} = \text{从 } i \text{ 点 } r \text{ 步到 } j \text{ 点的路径数目},$$

$$i, j = 1, 2, \cdots, n.$$

数学归纳法可给出证明，这里从略，读者自证之.

定理 1　令 $G_i^{(1)}$ 是有向图 $G = (V, E)$ 的 1- 因子子图(图 2-18)，L_i 表示 $G_i^{(1)}$ 的有向回路数目，$i = 1, 2, \cdots, p$. A 是图 G 的邻接矩阵，则

$$\det A = (-1)^n \sum_{i=1}^{p} (-1)^{L_i}.$$

证明　由行列式的定义

$$\det A = \sum_{(j)} \varepsilon_{j_1 j_2 \cdots j_n} a_{1j_1} a_{2j_2} \cdots a_{nj_n}.$$

其中 $(j) = (j_1 j_2 \cdots j_n)$ 是 $1, 2, \cdots, n$ 的一个排列，$\varepsilon_{j_1 j_2 \cdots j_n} = \begin{cases} 1, & \text{若}(j)\text{ 是偶排列}, \\ -1, & \text{否则}. \end{cases}$

$\sum_{(j)}$ 是对所有的排列 (j) 求和.

每一非零项

$$a_{1j_1}a_{2j_2}\cdots a_{nj_n}$$

对应边$(1,j_1),(2,j_2),\cdots,(n,j_n)$构成的一1-因子子图.反过来每一1-因子子图对应一非零项

$$a_{1j_1}a_{2j_2}\cdots a_{nj_n}.$$

现在来确定符号.假定 C 是1-因子子图的回路,它包含有边

$$(i_1,i_2),(i_2,i_3),\cdots,(i_l,i_1),$$

这些边的始点形成一排列

$$i_1 i_2\cdots i_{l-1}i_l,$$

其终点形成一排列

$$i_2 i_3\cdots i_l i_1.$$

容易证明将 $i_2i_3\cdots i_li_1$ 改为 $i_1i_2\cdots i_{l-1}i_l$ 只需作 $l-1$ 次交换.

假定1-因子子图对应于 $j_1j_2\cdots j_n$,包含有 L 个有向回路,又设这 L 个回路的长度分别为 $l_1,l_2,\cdots,l_L$,则将排列 $j_1j_2\cdots j_n$ 转换成 $1\,2\cdots n$ 需要 $(l_1-1)+(l_2-1)+\cdots+(l_L-1)=l_1+l_2+\cdots+l_L-L=n-L$ 次,所以

$$\epsilon_{j_1j_2\cdots j_n}=(-1)^{n-L}=(-1)^{n+L},$$

$$\det\boldsymbol{A}=(-1)^n\sum_{i=1}^{p}(-1)^{L_i}.$$

对于有向图 $G=(V,E)$ 的边 (i,j),给以权 w_{ij},令邻接矩阵

$$\boldsymbol{A}=[a_{ij}]_{n\times n},$$

其中

$$a_{ij}=\begin{cases}w_{ij}, & 若(i,j)\in E,\\ 0, & 其它.\end{cases}$$

可得

定理2　设 A 是图 G 的带权的邻接矩阵,则

$$\det\boldsymbol{A}=(-1)^n\sum_i(-1)^{L_i}w(G_i^{(1)}),$$

其中 $n=|V|$,$G_i^{(1)}$ 是 G 图第 i 个1-因子子图,L_i 是 $G_i^{(1)}$ 的有向回路数目,$w(C_i)$ 是 $G_i^{(1)}$ 子图的权的乘积.

定理的证明比较复杂,这里从略.

例6　$\boldsymbol{A}=\begin{bmatrix}1&0&3&0&1\\0&1&2&0&0\\1&3&0&1&1\\2&0&1&0&2\\0&0&3&0&1\end{bmatrix}$.则 $G_C(A)$ 如图2-19所示.

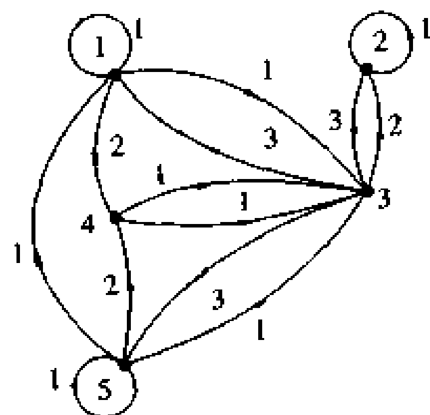
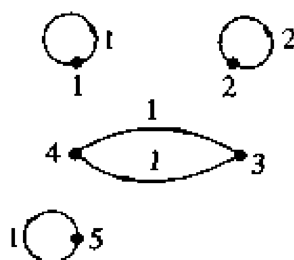


图 2-19

有

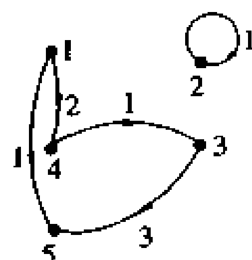
$G_1^{(1)}(A)$:



$L_1 = 4$

$w(G_1^{(1)}) = 1$

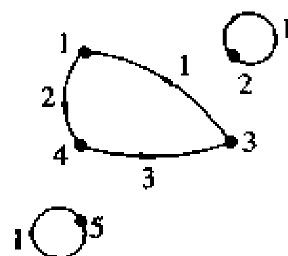
$G_2^{(1)}(A)$:



$L_2 = 2$

$w(G_2^{(1)}) = 6$

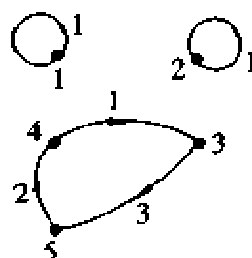
$G_3^{(1)}(A)$:



$L_3 = 3$

$w(G_3^{(1)}) = 6$

$G_4^{(1)}(A)$:



$L_4 = 3$

$w(G_4^{(1)}) = 6$

$$\det A = (-1)^5[(-1)^4\times 1+(-1)^2\times 6+(-1)^3\times 6+(-1)^3\times 6]$$
$$= -[1+6-6-6] = 5.$$

定义 1 设 $H_{ij}=(V,E^*)$ 是有向图 $G=(V,E)$ 的子图，$E^*\subsetneq E$. 若存在一条从 i 到 j 的有向路径 P_{ij}，以及与 P_{ij} 上的点无关且不相交的回路，则称图 H_{ij} 为有向图 G 关于连通 (i,j) 的 1- 因子子图

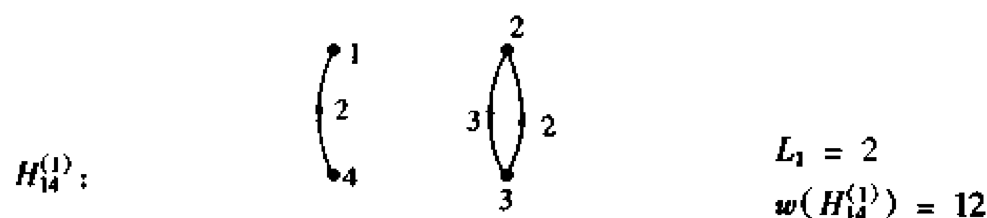
从方程组的行列式解法及行列式的科特斯图方法，可得

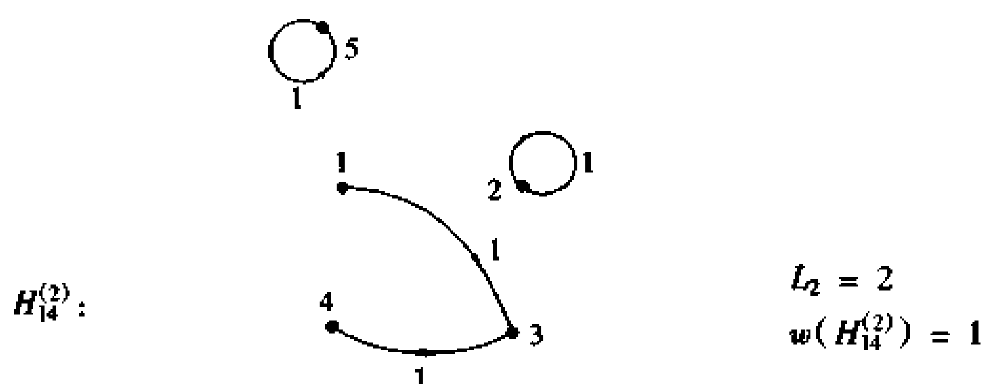
定理 3 若矩阵 A 的行列式不为零，则方程组 $AX=bx_{n+1}$ 的解为

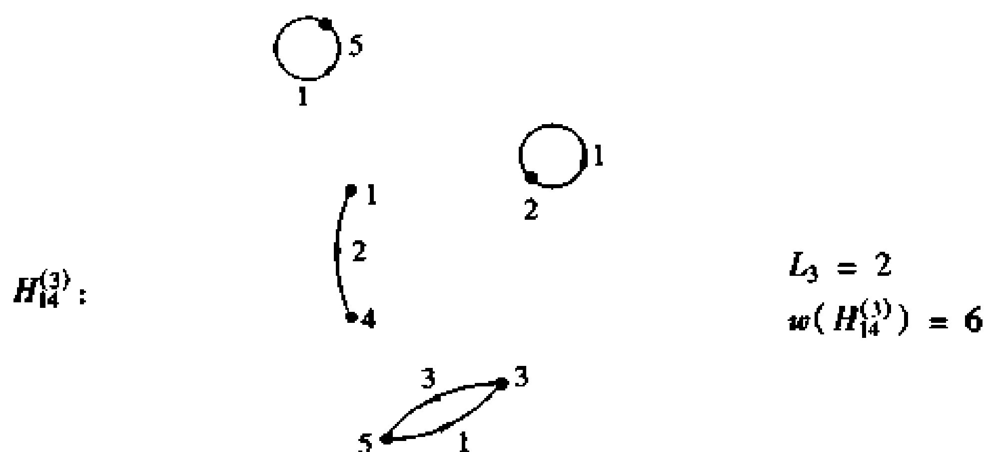
$$x_k = \frac{\sum_i (-1)_i^{L_i'} w(H_{n+1,k}^{(i)})}{\sum_j (-1)_j^{L_j} w(G_j^{(1)})} x_{n+1}, \quad k = 1,2,\cdots,n,$$

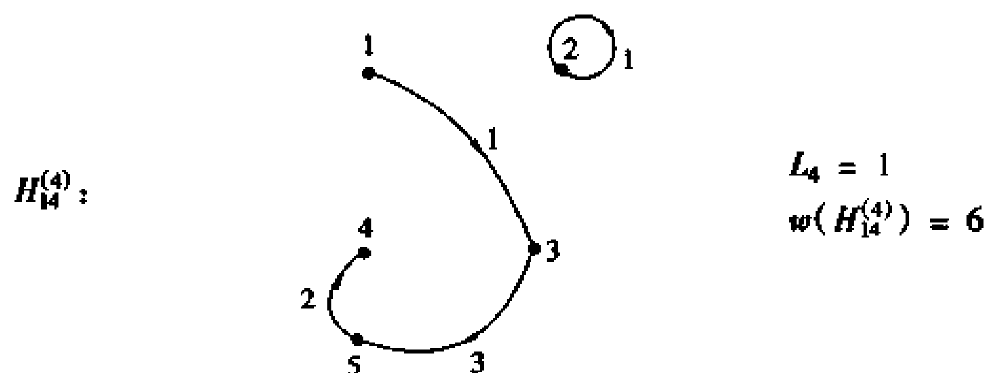
其中 $H_{n+1,k}^{(i)}$ 是 $H_{n+1,k}$ 的第 i 个 1-因子子图. L_i' 是图 $H_{n+1,k}^{(i)}$ 的有向回路的数目.

还是以图 2-19 为例,有









2.4 玛逊定理

如若方程组为

$$\begin{cases} x_k = (a_{kk}+1)x_k + \sum\limits_{i\neq k}^{n} a_{ki}x_i - b_k x_{n+1}, & k = 1,2,\cdots,n, \\ x_{n+1} = x_{n+1}, \end{cases}$$

或写为

$$(A^* + E_{n+1})X = X,$$

其中

$$X = (x_1, x_2, \cdots, x_n, x_{n+1})^{\mathrm{T}},$$
$$E_{n+1} = \mathrm{diag}(\underbrace{1,1,\cdots,1}_{n+1\text{个}}),$$

A^* 为前面介绍过的$(n+1)\times(n+1)$加边方阵.

$A^* + E_{n+1}$ 的科特斯图 $G_C(A^* + E_{n+1})$ 称为矩阵 A^* 的**玛逊信号流图**.

例 7

$$\begin{cases} 2x_1 + x_2 + x_3 = x_4, \\ x_1 - x_2 - 2x_3 = 0, \\ 2x_1 + x_2 + 2x_3 = -x_4, \end{cases} \tag{2-3}$$

或写成

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & -2 \\ 2 & 1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} x_4.$$

则

$$A^* = \begin{bmatrix} 2 & 1 & 1 & -1 \\ 1 & -1 & -2 & 0 \\ 2 & 1 & 2 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

或将方程组(2-3)改写为标准形式:

$$\begin{cases} x_1 = 3x_1 + x_2 + x_3 - x_4, \\ x_2 = x_1 \qquad\quad - 2x_3, \\ x_3 = 2x_1 + x_2 + 3x_3 + x_4, \\ x_4 = x_4. \end{cases}$$

$$X = (A^* + E_4)X,$$

其中

$$X = (x_1, x_2, x_3, x_4)^{\mathrm{T}},$$

$$A^* + E_4 = \begin{bmatrix} 3 & 1 & 1 & -1 \\ 1 & 0 & -2 & 0 \\ 2 & 1 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

图 2-20 是矩阵 A^* 的玛逊信号流图,是 $A^* + E_4$ 的科特斯信号流图.

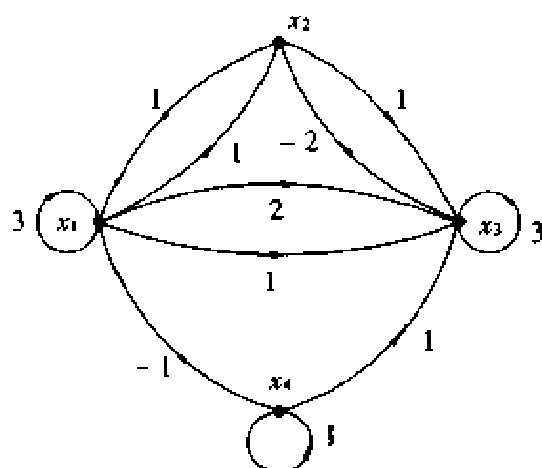


图 2-20

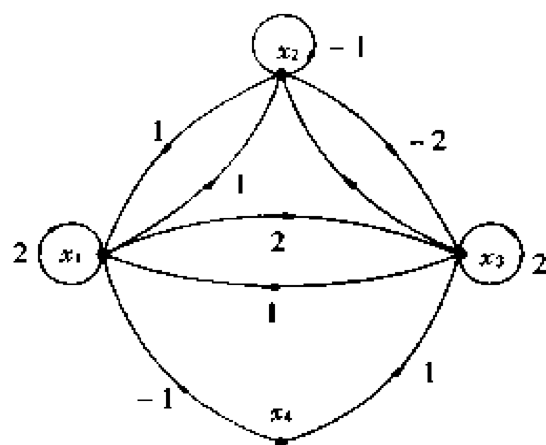


图 2-21

A^* 矩阵的玛逊信号流图用 $G_m(A^*)$ 表示它,实际上它表达了方程组 $AX = bx_{n+1}$. 它的每一个顶点代表一个变量. 对于某顶点 x_j,若有一权为 a_{ji} 的有向边从 x_i 引向 x_j,则对于 x_j 有一份 $a_{ji}x_i$ 的贡献. 所以 x_j 值等于所有 $x_i \to x_j$ 的有向边的权 a_{ji} 与 x_i 之积的和. 玛逊信号流图的这些性质与一些物理现象比较符合.

矩阵 A 的玛逊信号流图 $G_m(A)$ 和它的科特斯信号流图 $G_C(A)$ 的关系是:将 $G_C(A)$ 每个顶点的自环的权减去 1,无自环的顶点,则增加一权为 -1 的自环,便得 $G_m(A)$.

方程组(2-3)的 A^* 的科特斯信号流图 $G_C(A^*)$ 如图 2-21.

对应的玛逊信号流图为图 2-20.

关于玛逊信号流图有如下定理:

$$\det A = (-1)^n\left[1 - \sum_j w_{j1} + \sum_j w_{j2} - \sum_j w_{j3} + \cdots\right],$$

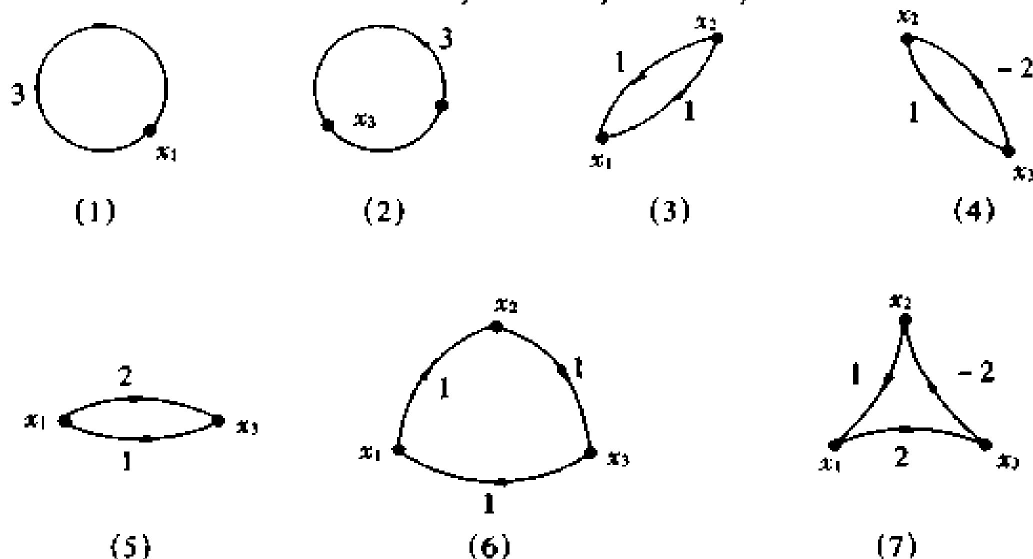


图 2-22

其中 w_{jk} 为图 $G_m(A)$ 的第 j 个子图的权之积，该子图由 k 个不相交的回路组成.

以例 7 说明之. $G_m(A)$ 图(图 2-20) 的回路见图 2-22，两两不相交的回路有图 2-23，不存在两个以上回路构成的子图. 所以

$$\begin{aligned}\det A &= (-1)^3[1-(3+3+1-2+2+1-4)+(9+3-6)]\\ &= -[1-4+6] = -3.\end{aligned}$$

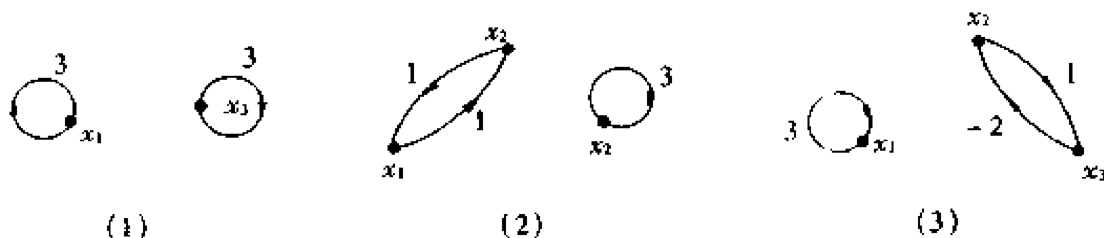


图 2-23

下面叙述玛逊定理.

定理 $G_m(A)$ 为方程组 $AX = bx_{n+1}$ 所对应的玛逊信号流图，A 为 $n\times n$ 非奇异矩阵，$C_1, C_2, \cdots, C_r$ 是图 $G_m(A)$ 的回路集合，w_i 是回路 C_i 上边的权的乘积，称为 C_i 的增益，$i = 1,2,\cdots,r$. 则有

$$\frac{x_h}{x_{n+1}} = \frac{N_h}{D}, \quad h = 1,2,\cdots,n,$$

其中

$$D = 1 - \sum_{i=1}^{r} w_i + \sum_{\substack{i,j\\ c_i, c_j\\ \text{不接触}}} w_i w_j - \sum_{\substack{i,j,k\\ c_i,c_j,c_k\\ \text{互不接触}}} w_i w_j w_k + \cdots + (-1)^a \sum_{\substack{i_1,i_2,i_a\\ c_{i_1},c_{i_2},\cdots,c_{i_a}\\ \text{互不接触}}} w_{i_1} w_{i_2} \cdots w_{i_a},$$

$$N_h = \frac{\sum_j w(P_{n+1,h}^{(j)})\Delta_j}{D}, \quad h = 1,2,\cdots,n.$$

又其中 $P_{n+1,h}^{(j)}$ 是从 x_{n+1} 到 x_h 的第 j 条道路，Δ_j 是从图 $G_m(A^*)$ 中不与 $P_{n+1,h}^{(j)}$ 相接触的点构成的子图的行列式.

回到例 7：

$$\begin{bmatrix} 2 & 1 & 1\\ 1 & -1 & -2\\ 2 & 1 & 2\end{bmatrix}\begin{bmatrix} x_1\\ x_2\\ x_3\end{bmatrix} = \begin{bmatrix} 1\\ 0\\ -1\end{bmatrix} x_4,$$

$G_m(A^*)$ 图如图 2-20.

$D = 3$. 以求 $\frac{x_2}{x_4}$ 为例.

$P_{42}^{(1)}: x_4 \to x_1 \to x_2, \qquad w(P_{42}^{(1)}) = -1,$

$P_{42}^{(2)}: x_4 \to x_1 \to x_3 \to x_2,\quad w(P_{42}^{(2)}) = 4,$

$P_{42}^{(3)}: x_4 \to x_3 \to x_2,\qquad w(P_{42}^{(3)}) = -2,$

$P_{42}^{(4)}: x_4 \to x_3 \to x_1 \to x_2,\quad w(P_{42}^{(4)}) = 1.$

$P_{42}^{(2)}$ 和 $P_{42}^{(4)}$ 不存在与其不相交的顶点的回路，而 $P_{42}^{(1)}$ 和 $P_{42}^{(3)}$ 则分别存在 x_3 点和 x_1 点的自环，权为 1. 所以

$$\Delta_1 = 1 - 3 = -2,\quad \Delta_2 = \Delta_4 = 1,\quad \Delta_3 = 1 - 3 = -2.$$

$$P_{42}^{(1)}\Delta_1 + P_{42}^{(2)}\Delta_2 + P_{42}^{(3)}\Delta_3 + P_{42}^{(4)}\Delta_4 = 2 + 4 + 4 + 1 = 11.$$

所以
$$\frac{x_2}{x_4} = \frac{11}{3}.$$

例 8　已知方程组

$$\begin{cases} x_1 - x_2 + x_3 = 0, \\ 2x_1 + 2x_2 + x_3 = 0, \\ x_1 + x_2 - x_3 = -b, \end{cases}$$

将这方程改写成为

$$\begin{cases} x_1 = x_2 - x_3, \\ x_2 = 2x_1 + 3x_2 + x_3, \\ x_3 = x_1 + x_2 + b, \end{cases}$$

或矩阵形式
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \\ 2 & 3 & 1 \\ 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} b.$$

该方程的玛逊信号流图如图 2-24(a)，回路有 6 个，如图 2-24(b) 所示.

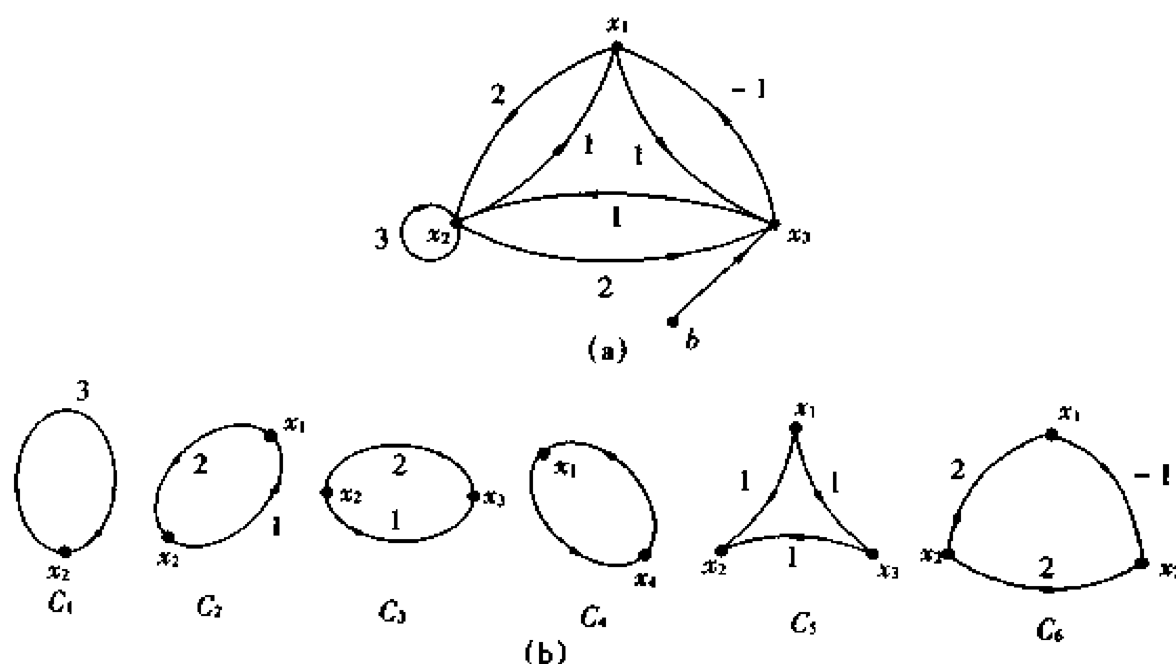


图 2-24

$C_1: x_2$ 自环，　　$C_2: x_1 \to x_2 \to x_1$，

$C_3: x_2 \to x_3 \to x_2$，　　$C_4: x_1 \to x_3 \to x_1$，

$C_5: x_1 \to x_3 \to x_2 \to x_1$，　　$C_6: x_1 \to x_2 \to x_3 \to x_1$，

其中 C_1 和 C_4 两两不相交. 所以

$$D = 1 - [-1 + 1 + 2 + 3 + 1 - 2] + (-1)3 = -6.$$

从 b 到 x_1 的道路有两条：

$P_{b1}^{(1)}: b \to x_3 \to x_2 \to x_1$，　　$w(P_{b1}^{(1)}) = 1$，

$P_{b2}^{(2)}: b \to x_3 \to x_1$，　　$w(P_{b1}^{(2)}) = -1$.

$P_{b1}^{(1)}$ 不存在与它不相交的回路，$\Delta_1 = 1$，

$P_{b1}^{(2)}$ 与 C_1 不相交，$\Delta_2 = 1 - 3 = -2$，

所以

$$x_1 = -\frac{1}{6}[1 + (-1)(-2)]b$$

$$= -\frac{1}{2}b.$$

例 9
$$\begin{bmatrix} a_{11} & 0 & a_{13} & 0 & a_{15} \\ 0 & a_{22} & a_{23} & 0 & 0 \\ a_{31} & a_{32} & 0 & a_{34} & a_{35} \\ a_{41} & 0 & a_{43} & 0 & a_{45} \\ 0 & 0 & a_{53} & 0 & a_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \\ 0 \\ 0 \\ b_5 \end{bmatrix}.$$

$G_m(A^*)$ 图如图 2-25 所示.

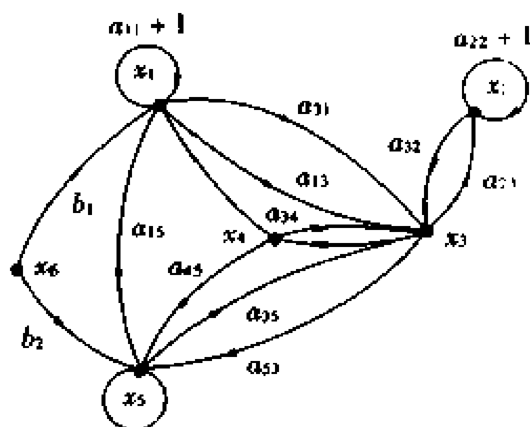


图 2-25

1- 因子子图：

$C_1: x_1$ 点自环，x_2 点自环，x_5 点自环，$x_3 \to x_4 \to x_3$，

$C_2: x_2$ 自环，x_5 自环，$x_7 \to x_4 \to x_3 \to x_1$，

$C_3: x_2$ 自环，$x_1 \to x_4 \to x_3 \to x_5 \to x_1$，

$C_4: x_1$ 自环，x_2 自环，$x_4 \to x_3 \to x_5 \to x_4$.

x_5 到 x_4 点的 1- 因子连通子图:

$$x_2 \to x_2,\quad x_6 \to x_5 \to x_1 \to x_3 \to x_4,$$
$$x_2 \to x_3 \to x_2,\quad x_6 \to x_5 \to x_1 \to x_4,$$
$$x_1 \to x_1,\quad x_2 \to x_3 \to x_2,\quad x_6 \to x_5 \to x_4,$$
$$x_2 \to x_2,\quad x_1 \to x_3 \to x_1,\quad x_6 \to x_5 \to x_4,$$
$$x_1 \to x_1,\quad x_2 \to x_2,\quad x_6 \to x_5 \to x_3 \to x_4,$$
$$x_2 \to x_2,\quad x_6 \to x_5 \to x_3 \to x_1 \to x_4,$$
$$x_2 \to x_2,\quad x_3 \to x_5 \to x_3,\quad x_6 \to x_1 \to x_4,$$
$$x_2 \to x_2,\quad x_6 \to x_1 \to x_3 \to x_5 \to x_4,$$
$$x_2 \to x_2,\quad x_5 \to x_5,\quad x_6 \to x_1 \to x_3 \to x_4,$$
$$x_5 \to x_5,\quad x_2 \to x_3 \to x_2,\quad x_6 \to x_1 \to x_4.$$

代入玛逊公式可得

$$\frac{x_4}{x_6} = \frac{N}{D},$$

其中

$$N = -b_1(a_{41}a_{35}a_{53}a_{22} + a_{31}a_{53}a_{45}a_{22} + a_{31}a_{43}a_{22}a_{55} + a_{41}a_{35}a_{53}a_{22})$$
$$- b_5(a_{15}a_{31}a_{43}a_{22} + a_{15}a_{41}a_{32}a_{23} + a_{35}a_{13}a_{41}a_{22} + a_{35}a_{43}a_{11}a_{22}),$$
$$D = a_{11}a_{22}a_{34}a_{43} + a_{55} + a_{22}a_{41}a_{34}a_{53}a_{15} -$$
$$a_{22}a_{41}a_{34}a_{13}a_{55} - a_{11}a_{22}a_{34}a_{45}a_{53}.$$

例 10　利用玛逊公式处理2.2节的例4(图 2-26).

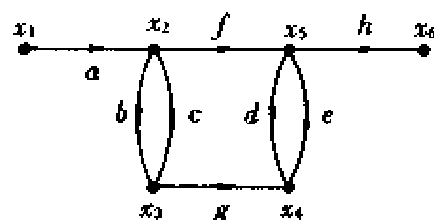


图 2-26

有回路如图 2-27 所示.

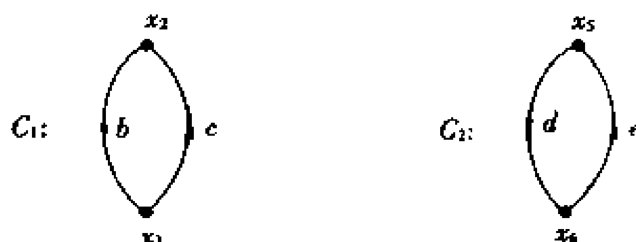
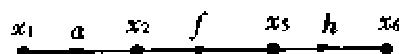
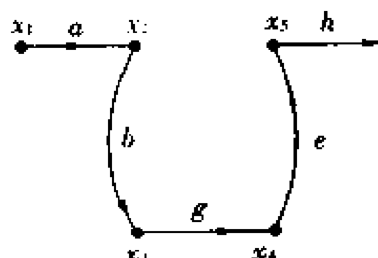


图 2-27

$P_{1b}^{(1)}$:



$P_{1b}^{(2)}$:



所以

$$\frac{x_6}{x_1}=\frac{afg+abgeh}{1-(bc+de)+bcde}.$$

参考文献

1 WAI-KAI CHEN. Applied graph theory, graphs and electrical networks, New York: North-Holland Publiching Company, 1976.

2 SWAMY M N S, Jhulasiraman K. Graphs, networks, and algorithms. New York: John Wiley d Sons, snc, 1981.

3 卢开隆,卢华明.图论及其应用.第2版.北京:清华大学出版社,1995.

·计算机数学卷·

第 11 篇

随机算法

编　者　张立昂
审校者　耿素云

目　　录

引　言

随机算法(randomized algorithms)又称作**概率算法**(probabilistic algorithms),是带随机操作的一类算法的总称.随机算法在计算过程中要产生符合规定要求的随机数,并根据产生的随机数决定后面的计算.例如,在某一步有两种选择:执行 A 或执行 B.此时随机地产生一个 0 或 1.若产生 1 则执行 A;若产生 0 则执行 B.这相当于投掷一枚硬币,并根据出现硬币的正面或反面决定做什么.把随机算法用于组合问题始于 20 世纪 70 年代,早期最著名的例子是索罗维(R.Solovay)、斯特拉逊(V.Strassen)和拉宾(M.O.Rabin)的素数判定随机算法.用拉宾算法测试梅森(Mersenne)数,即形如 2^p-1 的数(其中 p 是素数).取算法的错误概率不超过千分之一,对所有 $p<500$ 的梅森数进行测试,其结果与已知的结论完全一致,没有发生一次错误.运用拉宾算法还发现小于 2^{400} 的最大素数是 $2^{400}-593$.这一结论经 100 次以上的测试而不变,因而实际上可以认为 $2^{400}-593$ 是素数.随机算法的主要优点是简单和快速,现在在数据结构、排序与检索、数论与代数、图论、计算几何、并行计算等领域内都有广泛的应用.

在实际执行算法时所需要的随机数是由伪随机数发生器产生的.计算机都有伪随机数发生器,它产生伪随机数,而不是真正的随机数.实践表明,随机算法使用伪随机数能够工作得很好.但是,用伪随机数代替随机数工作能够好到什么程度,在理论上是一个尚未解决的问题.

1　随机算法的计算模型

1.1　蒙特卡罗法和拉斯维加斯法

随机算法在运行过程中要产生随机数,并根据产生的随机数决定后面的操作.因而,随机算法的运行带有随机性,不仅计算过程有随机性,计算结果也可能有随机性.对于同一个输入实例多次运行算法,每次的运行时间可能不同,计算结果也可能不同,甚至有时得不到结果.

可能给出错误结果的随机算法称作**蒙特卡罗法**(Monte Carlo algorithms).对于判定问题(只有两个可能的答案:"是"或"否"的问题),蒙特卡罗法又可以分成两类:单侧错误的和双侧错误的.如果答案为"是"和"否"时算法的错误概率都大于零,则是**双侧错误的**;如果在一种情况时算法的错误概率等于零,则是**单侧错误的**.

给出的结果总是正确的,但可能不给出结果的算法称作**拉斯维加斯法**(Las Vegas algorithms).

对于随机算法必须分析它的统计性能，如期望运行时间、输出错误结果或得不出结果的概率.

例1 随机快速排序(randomized quick sort).

快速排序 QS(Quick Sort)是常用的排序算法.它在待排序的数中指定一个数 y 作为划分点，把其余的数划分成两组 X_1 和 X_2，分别由小于和大于 y 的数组成.然后分别对 X_1 和 X_2 递归地应用算法.最后把排好序的 X_1 和 X_2 按照 X_1, y, X_2 的顺序排列.

随机快速排序(Rand QS)与 QS 的唯一区别是：不是"指定"划分点，而是改为从待排序的数中"等可能地随机取"一个数作为划分点.这是一个拉斯维加斯法，计算 n 个数的排序的期望运行时间为 $O(n\operatorname{lb} n)$.

例2 多项式恒等判定.

任给数域 F 上两个 n 次多项式 $f(x)$ 和 $g(x)$，问 $f(x) \equiv g(x)$?

把 $f(x)$ 和 $g(x)$ 都化成形如 $a_0x^n + a_1x^{n-1} + \cdots + a_n$ 的标准形式，问题就解决了.但是这种化简并不总是很简单的.例如，$f(x)$ 或 $g(x)$ 可能是含 x 的行列式.

现在采用另一种方式解决这个问题.如果 $f(x) \equiv g(x)$，则对任意的 $a \in F$ 有 $f(a) = g(a)$.因此，任取一个 $a \in F$，如果 $f(a) \neq g(a)$，则可以断言 $f(x) \not\equiv g(x)$.但是，当 $f(a) = g(a)$ 时不一定有 $f(x) \equiv g(x)$，因为 a 可能恰好是方程 $f(x) = g(x)$ 的根.设 $S \subseteq F$ 且 $|S| = 2n$，a 在 S 上服从均匀分布.由于当 $f(x) \not\equiv g(x)$ 时方程 $f(x) = g(x)$ 至多有 n 个不同的根，故 $f(a) = g(a)$ 的概率不超过 $n/|S| = 1/2$.

算法 Polyiden:

步1 从 S 中等可能地取一个数 a，这里 $S \subseteq F$ 且 $|S| = 2n$.

步2 if $f(a) \neq g(a)$ then return "不恒等"

else return "恒等"

当 $f(x) \equiv g(x)$ 时，算法总输出"恒等"；当 $f(x) \not\equiv g(x)$ 时，输出"恒等"的概率(即错误概率)不超过 1/2.这是单侧错误的蒙特卡罗法.

在实际运用中，1/2 的错误概率太大了，通过重复运行可以使错误概率任意地小.独立重复运行 t 次，即独立地产生 t 个随机数 $a_1, a_2, \cdots, a_t$.只要有一个 a_i 使 $f(a_i) \neq g(a_i)$，则必有 $f(x) \not\equiv g(x)$，输出"不恒等"；只有当所有 a_i 都有 $f(a_i) = g(a_i)(1 \leqslant i \leqslant t)$ 时，才输出"恒等".当 $f(x) \not\equiv g(x)$ 时，输出"恒等"的概率(即错误概率)不超过 $1/2^t$.取 $t = 10$，这个值小于 0.001，在实际运用中，概率如此小的随机事件几乎是不会发生的.

1.2 概率图灵机和概率计算复杂性类

1.2.1 概率图灵机

概率图灵机(probabilistic Turing machine，简记作 **PTM**)是随机算法的模型.PTM M 是一台总停机的非确定型图灵机，且作下述约定：

1° 每一个非停机格局恰好有 2 个后继.

2° 对每一个输入 x,M 的所有计算长度相同.

在上述约定下,M 关于输入 x 的计算可表成一棵完全正则二叉树,树根是初始格局,每一片树叶是一个停机格局.当停机在接受状态时,给这片树叶标记"是";当停机在拒绝状态时,给这片树叶标记"否".对 PTM M 的计算作如下解释:在计算的每一非停机步掷一枚硬币,根据所掷硬币的结果决定选择左儿子还是选择右儿子.硬币是均匀的,出现正面和反面的概率各为 1/2,并且各次掷币是相互独立的.计算到达每一片树叶的可能性相同.在全部树叶中,标记"是"的树叶所占的比例记作 $\alpha(M,x)$;标记"否"的树叶的比例记作 $\beta(M,x)$. $\alpha(M,x)$ 是 M 接受 x 的概率,$\beta(M,x)$ 是 M 拒绝 x 的概率. $\alpha(M,x)+\beta(M,x)=1$.

规定:当 $\alpha(M,x)>1/2$ 时,M 接受 x;当 $\beta(M,x)\geq 1/2$ 时,M 拒绝 x.

M 接受的所有字符串称作 M 接受的语言,记作 $L(M)$.

当 $x\in L(M)$ 时,M 的错误概率(拒绝 x 的概率)等于 $\beta(M,x)$;当 $x\notin L(M)$ 时,M 的错误概率(接受 x 的概率)等于 $\alpha(M,x)$. M 关于 x 的错误概率记作 $\mathrm{err}(M,x)$.

1.2.2 PP 机与 PP 类

定义 1 设一 PTM M,如果存在多项式 $p(n)$,使得对于每一个输入 x,M 的计算在 $p(|x|)$ 步内停机,则称 M 是一台**多项式时间概率图灵机**,简称作 **PP 机**(probabilistic polynomial-time TM).

所有 PP 机接受的语言组成的语言类记作 PP.

例 3 1/2-SAT.

问题:任给一个布尔表达式 f,回答"是"当且仅当在所有的真假值赋值中满足 f 的赋值的比例大于 1/2.

显然,1/2-SAT $\in$ PP.

1.2.3 BPP 机与 BPP 类

定义 2 满足下述条件的 PP 机 M 称作**错误有界的多项式时间概率图灵机**,简称作 **BPP 机**(bounded-error probabilistic polynomial-time TM):存在 $\epsilon\in(0,1/2)$,使得对所有的输入 x,要么 $\alpha(M,x)>1/2+\epsilon$,要么 $\beta(M,x)>1/2+\epsilon$.

所有 BPP 机接受的语言组成的语言类记作 BPP.

设 BPP 机 M,根据定义,当 $x\in L(M)$ 时 $\alpha(M,x)>1/2+\epsilon$;当 $x\notin L(M)$ 时 $\beta(M,x)>1/2+\epsilon$. 恒有 $\mathrm{err}(M,x)<1/2-\epsilon$.

通过独立重复运行,可以减少错误概率.有下述定理.

定理 1 设 $L\in$ BPP,$q(n)$ 是一个多项式,则存在 BPP 机 M 使得 $L(M)=L$,并且当 $x\in L$ 时 $\alpha(M,x)>1-2^{-q(|x|)}$,当 $x\notin L$ 时 $\beta(M,x)>1-2^{-q(|x|)}$.

根据定理 1-1,设 $L\in$ BPP,对任意的 $\delta>0$,存在 BPP 机 M 使得对所有的输入 x,$\mathrm{err}(M,x)<\delta$.

1.2.4 RP机与RP类

定义3 满足下述条件的PP机M称作**单侧错误有界的多项式时间概率图灵机**，简称作**RP机**(randomized polynomial-time TM)：对每一个输入x，要么$\alpha(M,x)>1/2$，要么$\beta(M,x)=1$.

所有RP机接受的语言组成的语言类记作RP.

设RP机M，根据定义，当$x\in L(M)$时$\alpha(M,x)>1/2$，此时$err(M,x)=\beta(M,x)<1/2$；当$x\notin L(M)$时$\beta(M,x)=1$，此时$err(M,x)=\alpha(M,x)=0$. RP机是单侧错误的且错误概率有界.

和BPP类似，设$L\in RP$，对任意的$\delta>0$，存在RP机M使得$L(M)=L$，并且当$x\in L$时$err(M,x)<\delta$，当$x\notin L$时$err(M,x)=0$.

BPP机和RP机是蒙特卡罗法的计算模型. BPP机是双侧错误的，RP机是单侧错误的.

例2中的多项式恒等判定问题属于RP.

1.2.5 ZPP机与ZPP类

对PP机做如下修改：M有一个特殊的状态$q_?$，当M对输入x的计算停机在状态$q_?$时，M既不接受x也不拒绝x. 称这样的计算是一个无效计算. 仍把这样的PP机称作PP机. 对于这样的PP机M，只有$\alpha(M,x)+\beta(M,x)\leqslant 1$. 无效计算的概率为$1-\alpha(M,x)-\beta(M,x)$.

定义4 满足下述条件的PP机M称作**零错误的多项式时间概率图灵机**，简称作**ZPP机**(zero-error probabilistic polynomial-time TM)：对所有的输入x，要么$\alpha(M,x)>1/2$且$\beta(M,x)=0$，要么$\beta(M,x)>1/2$且$\alpha(M,x)=0$.

所有ZPP机接受的语言组成的语言类记作ZPP.

根据定义，ZPP机的错误概率恒为0，不给出结果的概率小于1/2. 和BPP机、RP机一样，通过独立重复运行可以把不给出结果的概率减小到足够小. ZPP机是拉斯维加斯法的模型.

这几个语言类和P(多项式时间可接受的语言类)、NP(非确定型多项式时间可接受的语言类)、PSPACE(多项式空间可接受的语言类)以及它们的补之间有下述关系.

定理2 下述命题成立：

1° $P\subseteq ZPP$.

2° $ZPP=RP\cap co\text{-}RP$.

3° $RP\cup co\text{-}RP\subseteq BPP$.

4° $RP\subseteq NP$, $co\text{-}RP\subseteq co\text{-}NP$.

5° $BPP\subseteq PP$.

6° $NP\cup co\text{-}NP\subseteq PP$.

7° $PP\subseteq PSPACE$.

2 指 纹 术

2.1 指 纹 术

指纹术(fingerprinting technique)是一项重要的随机化技术.设全域 U 是一个很大的集合,任给 x、$y \in U$,要检查 x 与 y 是否相等.这个问题的确定算法的复杂度通常与 $|U|$ 有关.为了降低算法的复杂度,在 U 中取一个小得多的子集 V,作映射 F: $U \to V$ 满足下述条件:当且仅当 $F(x) = F(y)$ 时以足够大的概率保证 $x = y$.由于 V 比 U 小得多,检查 $F(x) = F(y)$ 要比检查 $x = y$ 容易得多.F 称作**指纹函数**,$F(x)$ 称作 x 的**指纹**.1.1 节中的例 1-2 就采用了指纹术.

例 1 任给3个 n 阶矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$,问:$\boldsymbol{AB} = \boldsymbol{C}$?

采用常规的确定算法需要计算 2 个 n 阶矩阵的乘积,已知的最快算法需要 $O(n^{2.376})$ 次算术运算.采用指纹术,随机地取一个 n 维 0-1 向量 $\boldsymbol{r}$,定义指纹函数 $F_r(\boldsymbol{A}) = \boldsymbol{Ar}$.当 $\boldsymbol{A} = \boldsymbol{B}$ 时,$F_r(\boldsymbol{A}) = F_r(\boldsymbol{B})$;当 $\boldsymbol{A} \neq \boldsymbol{B}$ 时,$F_r(\boldsymbol{A}) \neq F_r(\boldsymbol{B})$ 的概率 $\geqslant 1/2$.考虑下述算法:

步 1 随机地取 $\boldsymbol{r} \in \{0,1\}^n$,$\boldsymbol{r}$ 服从 $\{0,1\}^n$ 上的均匀分布.

步 2 计算 $\boldsymbol{x} = \boldsymbol{Br}$,$\boldsymbol{y} = \boldsymbol{Ax}$,$\boldsymbol{z} = \boldsymbol{Cr}$.

步 3 if y ≠ z then return "$\boldsymbol{AB} \neq \boldsymbol{C}$"
  else return "$\boldsymbol{AB} = \boldsymbol{C}$".

算法的运行时间为 $O(n^2)$.当 $\boldsymbol{AB} = \boldsymbol{C}$ 时,错误概率为零;当 $\boldsymbol{AB} \neq \boldsymbol{C}$ 时,错误概率 $\leqslant 1/2$.这是一个单侧错误的蒙特卡罗法.

例 2 检查字符串相等.

任给字母表 Σ 上的两个长度为 n 的字符串 a 和 b,问:$a = b$?

不妨设 $\Sigma = \{0,1,\cdots,d-1\}$,给定界限 $\tau = n^2 \mathrm{lb} n$,随机地取一个素数 $p \leqslant \tau$.把 Σ 上的字符串 $x = x_1 x_2 \cdots x_n$ 看作一个 d 进制数:$x = \sum_{i=1}^{n} x_i d^{i-1}$.

定义指纹函数

$$F_p(x) = x \pmod{p}, \tag{2-1}$$

这里 $x(\bmod p)$ 表示 x 除以 p 的余数.当 $a = b$ 时,$F_p(a) = F_p(b)$;当 $a \neq b$ 时,$F_p(a) = F_p(b)$ 的概率不超过 $n\mathrm{lb}d/\pi(\tau) = O(1/n)$,其中 $\pi(\tau)$ 是欧拉(Euler)函数,等于不超过 τ 的素数的个数.使用这个指纹函数容易设计出检查字符串相等的单侧错误的蒙特卡罗法,其错误概率为 $O(1/n)$.

2.2 多项式恒等检测

定理 1 设 $f(x_1,\cdots,x_n)$ 是数域 F 上的 n 元 d 次多项式.给定有穷子集 $S \subseteq F$.

随机变量 $a_1, a_2, \cdots, a_n$ 相互独立且都在 S 上服从均匀分布，则

$$P_r\{f(a_1, \cdots, a_n) = 0 \mid f(x_1, \cdots, x_n) \neq 0\} \leq \frac{d}{|S|}.$$

根据此定理，用指纹术容易设计出检测 n 元多项式恒等于0的单侧错误的蒙特卡罗法. 这是对1.1节中例2的推广.

2.3 模式匹配

设字母表 Σ, $X = x_1x_2\cdots x_n$ 和 $Y = y_1y_2\cdots y_m$ 是 Σ 上的两个字符串，$m \leq n$. X 称作正文(text)，Y 称作模式(pattern). 如果存在 $j(1 \leq j \leq n - m + 1)$ 使得对所有的 $1 \leq i \leq m$, $x_{j+i-1} = y_i$，则称模式 Y 在正文 X 中出现. 找到模式在正文中的出现(如果有的话)称作**模式匹配问题**(pattern matching problem).

记 $X(j) = x_jx_{j+1}\cdots x_{j+m-1}$，它是 X 中从第 j 位开始长度为 m 的子串. 采取2.1节例2中(2-1)式定义的指纹函数 F_p，对 $j = 1, 2, \cdots, n - m + 1$ 比较 $F_p(X(j))$ 和 $F_p(Y)$. 若 $F_p(X(j)) \neq F_p(Y)$，则 $X(j) \neq Y$，继续往下进行；若 $F_p(X(j)) = F_p(Y)$，为了确保匹配正确，再直接检查 $X(j) = Y$.

算法 KR[①]：

输入：正文 $X = x_1x_2\cdots x_n$ 和模式 $Y = y_1y_2\cdots y_m$, $m \leq n$.

输出：使 $X(j) = Y$ 的最小的 j(如果有的话).

步1　随机地取一个素数 $p \leq n^2 m \mathrm{lb}(n^2 m)$.

步2　for $j = 1, 2, \cdots, n - m + 1$ do

　　if $F_p(X(j)) = F_p(Y)$ then

　　　if $X(j) = Y$ then return j

步3　return "no"

算法KR是一个拉斯维加斯法，并总能给出结果. 当 Y 不在 X 中出现时，算法输出"no". 由于

$$X(j+1) = d(X(j) - d^{m-1}x_j) + x_{j+m},$$

有递推公式

$$F_p(X(j+1)) = (d(F_p(X(j)) - a x_j) + x_{j+m})(\mathrm{mod}\, p),$$

其中 $a = d^{m-1}(\mathrm{mod}\, p)$.

在步骤2，除 $j = 1$ 之外，用上述递推公式可在常数时间内计算每一个 $F_p(X(j))$. 因为当 $X(j) \neq Y$ 时 $F_p(X(j)) = F_p(Y)$ 的概率很小，所以很少出现在 $X(j) \neq Y$ 的情况下要检查 $X(j) = Y$. 算法的期望运行时间为 $O(n + m)$. 这里假定关于 $O(\mathrm{lb}\, n)$ 位数的运算(如检查 $F_p(X(j)) = F_p(Y)$)可在常数时间内完成，而关于 m 位数的运算需要时间 $O(m)$.

---

① 算法KR是由卡普(R. M. Karp)和拉宾于1987年提出的.

3 素数判定

3.1 预备知识

a 可以整除 b，记作 $a \mid b$. a 和 b 的最大公因数记作 $\gcd(a,b)$. a 和 b 互素当且仅当 $\gcd(a,b)=1$. 设 $n>0$，如果用 n 除 a 和 b，余数相同，则称 a 和 b 模 n 同余，记作 $a \equiv b \pmod{n}$. 记

$$\Phi(n)=\{x \mid 0 \leqslant x \leqslant n-1 \text{ 且 } x \text{ 与 } n \text{ 互素}\}.$$

对于素数 p，$\Phi(p)=\{0,1,\cdots,p-1\}$.

定理 1(费马(Fermat)小定理)　设 p 是素数，若 a 与 p 互素，则

$$a^{p-1} \equiv 1 \pmod{p}.$$

定义 1　设 $n>1$，a 与 n 互素. 如果存在 x 使

$$a \equiv x^2 \pmod{n},$$

则称 a 是模 n 的二次剩余，否则称 a 是模 n 的二次非剩余.

定理 2(欧拉准则)　设 p 为奇素数，a 与 p 互素，则当 a 是模 p 的二次剩余时，

$$a^{(p-1)/2} \equiv 1 \pmod{p};$$

当 a 是模 p 的二次非剩余时，

$$a^{(p-1)/2} \equiv -1 \pmod{p}.$$

定义 2　设 p 为奇素数，a 与 p 互素. 称

$$\left(\frac{a}{p}\right)=\begin{cases}1, & \text{若 } a \text{ 是模 } p \text{ 的二次剩余},\\ -1, & \text{若 } a \text{ 是模 } p \text{ 的二次非剩余}\end{cases}$$

为勒让德(Legendre)符号.

定义 3　设正奇数 $n=p_1p_2\cdots p_t$，$p_i(i=1,2,\cdots,t)$ 是素数，又设 a 与 n 互素. 称

$$\left(\frac{a}{n}\right)=\prod_{i=1}^{t}\left(\frac{a}{p_i}\right)$$

为雅可比(Jacobi)符号.

当 p 为奇素数，a 与 p 互素时，

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

定理 3　设 n 和 m 是正奇数且互素，a 与 n 互素，b 与 n 互素. 雅可比符号有下述性质：

1° $\left(\frac{ab}{n}\right)=\left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$.

2° 若 $a \equiv b \pmod{n}$，则 $\left(\frac{a}{n}\right)=\left(\frac{b}{n}\right)$.

3° $\left(\frac{m}{n}\right)\left(\frac{n}{m}\right)=(-1)^{(m-1)(n-1)/4}$.

4° $\left(\frac{1}{n}\right) = 1.$

5° $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8} = \begin{cases} 1, & 若 n \equiv 1 或 7 \pmod 8; \\ -1, & 若 n \equiv 3 或 5 \pmod 8. \end{cases}$

根据定理3,$\left(\frac{a}{n}\right)$是多项式时间可计算的.算法的思想如下例所示.

$$\left(\frac{182}{265}\right) = \left(\frac{2}{265}\right)\left(\frac{91}{265}\right) \qquad 182 = 2 \times 91,(性质 1°)$$

$$= \left(\frac{91}{265}\right) \qquad 265 \equiv 1 \pmod 8,(性质 5°)$$

$$= (-1)^{(91-1)(265-1)/4}\left(\frac{265}{91}\right) \qquad (性质 3°)$$

$$= \left(\frac{83}{91}\right) \qquad 265 \equiv 83 \pmod{91},(性质 2°)$$

$$= (-1)^{(83-1)(91-1)/4}\left(\frac{91}{83}\right) \qquad (性质 3°)$$

$$= -\left(\frac{2}{83}\right)^3 \qquad 91 \equiv 2^3 \pmod{83},(性质 2°)$$

$$= -(-1)^3 \qquad 83 \equiv 3 \pmod 8,(性质 5°)$$

$$= 1.$$

3.2　索罗维-斯特拉逊算法

素数判定是随机算法最成功的例子之一.至今还不清楚是否存在判定素数的多项式时间的确定算法.

根据欧拉准则(定理2),当 n 为奇素数时,对每一个 $1 \leqslant x \leqslant n-1$,都有

$$\left(\frac{x}{n}\right) \equiv x^{(n-1)/2} \pmod n.$$

设 n 是一个奇数,若存在 $x \in \Phi(n)$ 使得

$$\left(\frac{x}{n}\right) \not\equiv x^{(n-1)/2} \pmod n,$$

则 n 是合数.这样的 x 是 n 为合数的"见证".

记

$$J_n = \{x \mid x \in \Phi(x) 且 \left(\frac{x}{n}\right) \equiv x^{(n-1)/2} \pmod n\}.$$

当 n 为奇素数时,$J_n = \Phi(n)$;当 n 为奇合数时,$\Phi(n) - J_n$ 中的数都是 n 的合数"见证".

定理4　对于每一个奇合数 n,$|J_n| \leqslant \frac{1}{2}|\Phi(n)|$.

算法SS[①]

① 算法SS由索罗维和斯特拉逊于1977提出.

输入:整数 $n \geq 2$.
输出:"素数"或"合数".
步 1 if $n = 2$ then return"素数"
步 2 if $2 \mid n$ then return"合数"
步 3 等可能地从 $1,2,\cdots,n-1$ 中随机取一个数 x.
步 4 if $\gcd(x,n) \neq 1$ then return"合数".
步 5 if $\left(\frac{x}{n}\right) \equiv x^{(n-1)/2} \pmod{n}$ then return"素数"
else return"合数"

算法是多项式时间的.当 n 为素数时,算法总输出"素数".当 n 为合数时,算法可能给出错误的回答.由定理 4,错误概率不超过 1/2.这是关于合数判定的 RP 算法.

3.3 拉宾算法

拉宾于 1980 年给出另一个素数判定的随机算法.根据费马小定理,如果存在 $1 \leq x \leq n-1$ 使得 $x^{n-1} \not\equiv 1 \pmod{n}$,则 n 是合数.这样的 x 是 n 的合数见证.设 $1 \leq x \leq n-1$,若 $\gcd(x,n) \neq 1$,则 $x^{n-1} \not\equiv 1 \pmod{n}$.可惜对于某些合数,这样的见证所占的比例太小.再考查 $\Phi(n)$ 中的数,记

$$F_n = \{x \mid x \in \Phi(n) \text{ 且 } x^{n-1} \equiv 1 \pmod{n}\}.$$

当 n 为素数时,$F_n = \Phi(n)$;当 n 为合数时,$\Phi(n) - F_n$ 中的数都是 n 的合数见证.

定义 4 如果 n 是合数且 $F_n = \Phi(n)$,则称 n 是一个**卡米柴尔(R.D. Carmichael)数**.

最小的 5 个卡米柴尔数是:$561 = 3 \times 11 \times 17$, $1105 = 5 \times 13 \times 17$, $1729 = 7 \times 13 \times 19$, $2465 = 5 \times 17 \times 19$, $2821 = 7 \times 13 \times 31$.已经知道有无穷多个卡米柴尔数.

根据定义,当 n 为卡米柴尔数时,$\Phi(n)$ 中没有一个数符合上述合数见证的条件.因此,需要进一步寻找可以作为合数见证的条件.

拉宾用下述条件作为合数见证,记作 $W(x)$:

1° $1 \leq x \leq n-1$.

2° $x^{n-1} \not\equiv 1 \pmod{n}$ 或者存在 $i \geq 1$,使得 $2^i \mid (n-1)$ 且 $1 < \gcd(x^{(n-1)/2^i} - 1, n) < n$.

定理 5 如果 n 是合数且不是卡米柴尔数,则

$$|F_n| \leq \frac{1}{2} |\Phi(n)|.$$

定理 6 如果 n 是卡米柴尔数,则 $\Phi(n)$ 中符合条件 $W(x)$ 的数所占比例不小于 3/4.

算法 Rabin:
输入:整数 $n \geq 2$.
输出:"素数"或"合数"

步1　if $n=2$ then return “素数”
步2　if $2\mid n$ then return “合数”
步3　从 $1,2,\cdots,n-1$ 中等可能地随机取一个数 x
步4　if 符合条件 $W(x)$ then return“合数”else return“素数”

算法是多项式时间的.当 n 是素数时,一定输出“素数”.当 n 是合数时,根据定理5和定理6,错误概率不超过1/2.这也是关于合数判定的RP算法.

米勒(G.L.Miller)证明,假设广义黎曼(G.F.B.Reimann)猜想成立,则存在常数 c 使得当 n 为合数时,在 $[1,c\text{lb}^2 n]$ 中必有符合条件 $W(x)$ 的数.只要检查 $[1,c\text{lb}^2 n]$ 中的每一个数是否符合条件 $W(x)$ 就可以判定 n 是否是素数.从而得到素数判定的多项式时间的确定算法.

4　图论算法

4.1　最小割集

设 $G=(V,E)$ 是一个无向多重图,$|V|=n$,$|E|=m$.这里多重图是指两个顶点之间可以有若干条边,但是没有环(端点是同一个顶点的边).多重图也可以看作带正整数权的赋权图,边 $[u,v]$ 的权等于 u、v 之间的边数.设 $C\subsetneq V$,$\overline{C}=V-C$,由端点分别在 C 和 $\overline{C}$ 中的边组成的集合叫做 G 的割集,记作 $(C,\overline{C})$.G 的边数最少的割集叫做最小割集.**最小割集问题**:给定多重图 G,求 G 的最小割集.

下述操作称作收缩边 e:把 e 的端点 x 和 y 合并成一个顶点 w,删去连接 x 和 y 的边.对每一点 $z\in V-\{x,y\}$,把每一条连接 z 和 x、z 和 y 的边替换成一条连接 z 和 w 的边,图 G 经过收缩边 e 得到的图记作 G/e.见图4-1,其中(a)是图 G,(b)是 G/e.更一般地,设 $F\subseteq E$,收缩 F 中的边得到的图记作 G/F.

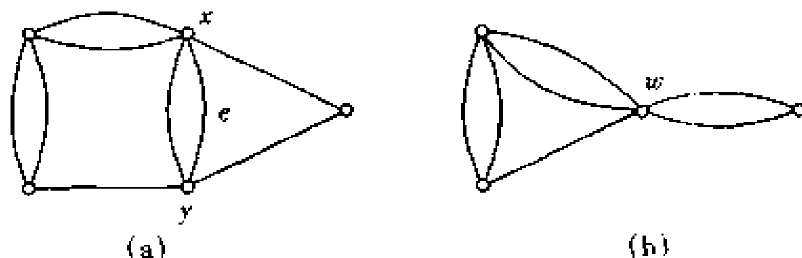


图4-1

给定多重图 G 和正整数 $t(2\leqslant t<n)$.每一次等可能地随机取一条边 e,收缩边 e,共进行 $n-t$ 次.把这个过程记作CONTRACT,所得到的图记作 H.H 的每一个顶点对应 G 的一个顶点子集,称作“超顶点”.H 有 t 个超顶点.当 $t=2$ 时,设 H 的两个超顶点分别对应 C 和 $\overline{C}$,则 H 的边集恰好是割集 $(C,\overline{C})$.

运算 CONTRACT 有下述性质：

(1) 设 $C \subsetneq V$，$(C,\overline{C})$ 是 H 的割集，当且仅当 $(C,\overline{C})$ 中的边没有被收缩.

(2) H 的最小割集的边数不小于 G 的最小割集的边数.

(3) 设 $(K,\overline{K})$ 是 G 的最小割集，$(K,\overline{K})$ 中没有边被收缩的概率不小于 $\left(\frac{t}{n}\right)^2$.

算法 MIN CUT[①]：

输入：多重图 $G = (V,E)$.

输出：$C \subsetneq V$.

步 1　令 $n \leftarrow |V|$.

步 2　if $n \leqslant 6$ then 用穷举法求得 G 的最小割集 $(C,\overline{C})$，return C.

步 3　令 $t \leftarrow \lceil 1 + n/\sqrt{2}\rceil$.

步 4　对 G 和 t 运用 2 次算法 CONTRACT，设分别得到 H_1 和 H_2.

步 5　对 H_1 和 H_2 递归地运用算法 MINCUT，设分别得 C_1 和 C_2.

步 6　if $|(C_1,\overline{C_1})| < |(C_2,\overline{C_2})|$ then $C \leftarrow C_1$ else $C \leftarrow C_2$.

步 7　return C

定理 1　算法 MINCUT 对 n 个顶点的多重图 G 的运行时间为 $O(n^2\mathrm{lb}n)$，使用空间为 $O(n^2)$，成功的找到 G 的最小割集的概率不小于 $\alpha/\mathrm{lb}n$，其中 $\alpha > 0$ 是一个常数.

由定理 1，对任意的 $d > 0$，存在正整数 c，独立地重复运行算法 $c\mathrm{lb}n$ 次，成功地找到最小割集的概率不小于 $1 - e^{-d}$，运行时间为 $O(n^2\mathrm{lb}^2 n)$. 这个时间比利用网络技术的确定算法的时间 $O(n^3)$ 少得多.

4.2　最小生成树

求最小生成树的确定算法的运行时间都是超线性的，本节介绍一个线性时间的随机算法.

设赋权无向图 $G = (V,E)$，$|V| = n$，$|E| = m$，边 e 的权为 $w(e)$. 这里不必假设 G 是连通的. 当 G 连通时，求 G 的最小生成树；当 G 不连通时，求 G 的最小生成森林. 下面假设边的权都不相同，从而 G 的最小生成树(森林)是唯一的. 这个假设不失一般性. 实际上，当某些边的权相等时，可以人为地指定它们的顺序，从而得到 E 上的全序关系.

4.2.1　波茹夫卡(O. Boruvka)操作

设 $e \in E$，像 4.1 节那样收缩 e，但这里在任意两点之间只保留一条权最小的边(如果有的话)，而删去其余的平行边. 把这称作短接 e.

波茹卡夫算法又称作短接算法，是经典的最小生成树算法. 它的基本做法是：

---

① 算法 MINCUT 由卡吉(D. R. Karger)和斯坦(C. Stein)提出.

对每一个顶点标出与它关联的权最小的边,短接所有带标记的边.把这个过程称作一次**波茹夫卡操作**.重复执行波茹夫卡操作,直到只剩下一个顶点为止.所有被短接的边组成 G 的最小生成树(森林).其算法的运行时间为 $O((m+n)\lg n)$.

4.2.2 随机图和重的边

设 F 是 G 的一个森林,u、$v \in V$.如果 u、v 在 F 的同一个连通分支内,则在 F 中存在连接 u 和 v 的唯一路径.把这条路径上的边的最大权记作 $w_F(u,v)$.如果 u、v 不在 F 的同一个连通分支内,则 $w_F(u,v)=\infty$.设边 $e=[u,v]$,如果 $w(e)>w_F(u,v)$,则称 e 是 F-重的,否则称 e 是 F-轻的.

定义1 设 $G=(V,E)$,$0<p<1$.随机图 $G(p)=(V,E(p))$ 的边规定如下:每一条边 $e\in E$ 属于 $E(p)$ 的概率为 p 且是相互独立的.

性质:

1° F-重的边不可能在最小生成树(森林)内.

2° 给定 G 的森林 F,可以在时间 $O(m+n)$ 内识别出 G 中所有 F-重的边.

3° 设 F 是随机图 $G(p)$ 的最小生成森林,则 G 中 F-轻的边数的期望不超过 n/p.

4.2.3 线性时间的最小生成树随机算法

算法 Rand MST①:

输入:赋权无向图 G.

输出:G 的最小生成森林 F.

步1 对 G 进行3次波茹夫卡操作,得到 G_1,C 为所有被短接的边.若 G_1 没有边,则 return $F=C$.

步2 令 $G_2=G_1(1/2)$.

步3 对 G_2 递归地运用算法 Rand MST,得到 F_2.

步4 删去 G_1 中所有 F_2-重的边,得到 G_3.

步5 对 G_3 递归地运用算法 Rand MST,得到 F_3.

步6 return $F=C\cup F_3$.

定理2 用算法 Rand MST 计算 n 个顶点、m 条边的图的最小生成森林的期望运行时间为 $O(m+n)$.

4.3 最大割集的随机近似算法

与求最小割集不同,求最大割集是NP难的.它不存在多项式时间的确定算法(除非 P = NP),甚至也不存在有效的随机算法(除非 NP $\subseteq$ BPP).容易给出最大割集的1/2-近似算法.虽然已经提出多个近似算法,但是近似性能都没有实质性的改进,或者没有能够证明有实质性的改进.歌门斯(M.X.Goemens)和威廉森(D.P.

① 算法 Rand MST 由卡吉·克莱因(P.N.Klein)和塔简(R.E.Tarjan)于1995年提出.

Williamson）于 1994 年提出一个随机近似算法，其平均性能比可以任意接近 $\alpha\approx 0.87856$，这里

$$\alpha=\min_{0<\theta\leqslant\pi}\left\{\frac{2}{\pi}\cdot\frac{\theta}{1-\cos\theta}\right\}. \tag{4-1}$$

算法的基本思想是把问题表述成整数二次规划，然后把这个整数二次规划松弛成一个半定规划．解半定规划，最后用一种巧妙的随机方法从这个半定规划的解得到原问题的解．这个方法具有相当的普遍意义，运用于最大可满足问题（MAX SAT）等也取得很好的效果．

设无向图 $G=(V,E)$，其中 $V=\{1,2,\cdots,n\}$．每一条边 $[i,j]\in E$ 有非负权 $w_{ij}=w_{ji}$．约定：对每一对 $1\leqslant i,j\leqslant n$，当 $[i,j]\notin E$ 时，$w_{ij}=w_{ji}=0$．设 $C\subseteq V$，令

$$y_i=\begin{cases}1, & i\in C,\\ -1, & i\notin C,\end{cases}\quad i=1,2,\cdots,n,$$

则

$$w(C,\overline{C})=\frac{1}{2}\sum_{i<j}w_{ij}(1-y_iy_j).$$

于是，最大割集问题可表成下述整数二次规划：

$$(\mathrm{Q})\quad\begin{cases}\max\ \dfrac{1}{2}\sum\limits_{i<j}w_{ij}(1-y_iy_j),\\ \text{s.t.}\quad y_i\in\{-1,1\},\ i=1,2,\cdots,n.\end{cases}$$

考虑（Q）的下述松弛形式：

$$(\mathrm{P})\quad\begin{cases}\max\ \dfrac{1}{2}\sum\limits_{i<j}w_{ij}(1-\boldsymbol{v}_i\cdot\boldsymbol{v}_j),\\ \text{s.t.}\quad \boldsymbol{v}_i\in S_n,\ i=1,2,\cdots,n,\end{cases}$$

其中 $S_n=\{(x_1,x_2,\cdots,x_n)\mid\sum_{i=1}^{n}x_i^2=1\}$ 是 n 维单位球面，$\boldsymbol{v}_i\cdot\boldsymbol{v}_j$ 是向量 $\boldsymbol{v}_i$ 与 $\boldsymbol{v}_j$ 的内积．

记 $\boldsymbol{B}$ 是以 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n$ 为列的矩阵．令 $\boldsymbol{Y}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$，$\boldsymbol{Y}$ 是 n 阶半正定对称矩阵，其元素 $y_{ij}=\boldsymbol{v}_i\cdot\boldsymbol{v}_j$ 且 $y_{ii}=1,\ i,j=1,2,\cdots,n$．于是，规划（P）可改写成

$$(\mathrm{SD})\quad\begin{cases}\max\ \dfrac{1}{2}\sum\limits_{i<j}w_{ij}(1-y_{ij}),\\ \text{s.t.}\quad y_{ii}=1,\ i=1,2,\cdots,n,\\ \qquad\ \boldsymbol{Y}=[y_{ij}]\text{ 是半正定对称矩阵}.\end{cases}$$

（SD）是一个半定规划（semidefinite programming）．任给 $\varepsilon>0$，用内点法可在问题的规模和 $\mathrm{lb}(1/\varepsilon)$ 的多项式时间内求得问题的近似解，其目标函数值不小于 $Z^*-\varepsilon$，这里 Z^* 是问题的最优值．

根据线性代数中的知识，每一个半正定对称矩阵 $\boldsymbol{Y}$ 都可表成 $\boldsymbol{Y}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$，并且可以用不完全的楚列斯基分解法（incomplete Cholesky decomposition）解得 $\boldsymbol{B}$，计算时间为 $O(n^3)$．因而，可以从（SD）的解得到（P）的解．

剩下的问题是如何从(P)的解 $v_1, v_2, \cdots, v_n$ 得到(Q)的解 $y_1, y_2, \cdots, y_n$ 或 $C \subseteq V$. 为此,引入一个单位长度的 n 维随机向量 r,它在 S_n 上服从均匀分布.取

$$C = \{i \mid v_i \cdot r \geqslant 0, i = 1, 2, \cdots, n\},$$

即以 r 为法向量的过原点的平面把 S_n 分成两部分,所有落在 r 指向一侧的 v_i 所对应的顶点 i 构成 C.

算法 GW:

输入:无向图 $G = (V, E)$ 及非负权 w_{ij}.

输出:$C \subseteq V$.

步1　解半定规划(SD),得到 Y.

步2　解 $B^{\mathrm{T}}B = Y$,得到 $B = (v_1, v_2, \cdots, v_n)$.

步3　随机地产生一个在 S_n 上服从均匀分布的 r.

步4　令 $C = \{i \mid v_i \cdot r \geqslant 0, i = 1, 2, \cdots, n\}$,输出 C.

定理3　对任意的 $\epsilon > 0$,算法 GW 在 n、$\mathrm{lb}(\sum\limits_{i<j} w_{ij})$ 和 $\mathrm{lb}(1/\epsilon)$ 的多项式时间内求得 C, $W = w(C, \overline{C})$ 的期望满足下述不等式:

$$E(W) \geqslant (\alpha - \epsilon) W_{MC}^*,$$

其中 W_{MC}^* 是 G 的最大割集的权,α 由(4-1)式给出.

5　几何算法

随机增长构造(randomized incremental costraction)是设计随机几何算法的常用有效方法.它的基本做法是:设问题中含有 n 个对象,算法分 n 步进行.按照随机的顺序,每一步考虑一个对象,考虑它对所要构造的几何对象的影响.本章介绍的几个算法都是用随机增长构造方法设计的.

多数随机几何算法与同一问题的确定算法有大体相同的运行时间,主要优点是简单.

5.1　平面上的凸包

定义1　设 S 是平面上的 n 个点组成的集合.平面上包括 S 的最小凸集称作 S 的凸包(convex hull),记作 $\mathrm{conv}(S)$.

$\mathrm{conv}(S)$ 是以 S 中的某些点为顶点的凸多边形,如图5-1所示.$\mathrm{conv}(S)$ 可以用边界上的顶点按逆时针方向的排列表示.

计算平面凸包的随机增长算法如下:

首先随机排列 S 中的点,设为 $p_1, p_2, \cdots, p_n$.记 $S_i = \{p_1, p_2, \cdots, p_i\}$.算法分 n 个阶段进行,依次构造出 $\mathrm{conv}(S_1), \mathrm{conv}(S_2), \cdots, \mathrm{conv}(S_n) = \mathrm{conv}(S)$.在阶段 i,把 p_i 添加到 $\mathrm{conv}(S_{i-1})$ 上构造出 $\mathrm{conv}(S_i)$.

取定 $\mathrm{conv}(S)$ 的一个内点,例如取 $\mathrm{conv}(S_3)$ 的中心 p_0(不妨设 p_1、p_2、p_3 不共

线，且三角形 $p_1p_2p_3$ 的中心不属于 S）. 用一个（循环的）链表存放 $\mathrm{conv}(S_i)$. 对每一个 $p_j(i < j \leqslant n)$，从 p_0 出发经过 p_j 的射线与 $\mathrm{conv}(S_i)$ 的一条边 η 相交，称点 p_j 切割边 η. 建立从 p_j 到 η 的双向指针. 于是，对 $\mathrm{conv}(S_i)$ 的每一条边可以枚举出所有切割这条边的点，使用的时间正比于这些点的个数.

图 5-1

在阶段 i，首先根据线段 $\overline{p_0p_i}$ 和 p_i 切割的 $\mathrm{conv}(S_{i-1})$ 的边，可以判断出 p_i 在 $\mathrm{conv}(S_{i-1})$ 的内部还是在 $\mathrm{conv}(S_{i-1})$ 的外部. 如果 p_i 在 $\mathrm{conv}(S_{i-1})$ 的内部，则 $\mathrm{conv}(S_i) = \mathrm{conv}(S_{i-1})$. 只需删去 p_i 的指针，进入下一个阶段. 如果 p_i 在 $\mathrm{conv}(S_{i-1})$ 的外部，则要修改存放凸包边界的链表和有关的指针. 此时，p_i 在 $\mathrm{conv}(S_i)$ 的边界上. $\mathrm{conv}(S_{i-1})$ 边界上的点可分成 3 类：

(1) 不在 $\mathrm{conv}(S_i)$ 的边界上，需要删去.

(2) 在 $\mathrm{conv}(S_i)$ 的边界上且与 p_i 相邻. 这样的点有两个，分别记作 v_1 和 v_2.

(3) 在 $\mathrm{conv}(S_i)$ 的边界上且不与 p_i 相邻. 对这些点不需做任何修改.

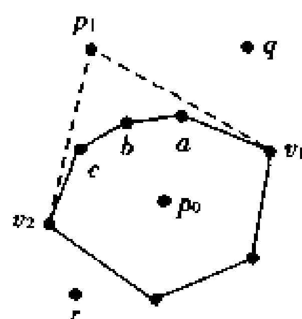


图 5-2

设直线段 $\overline{p_0p_i}$ 与 $\mathrm{conv}(S_{i-1})$ 的边 η 相交，则 η 的两个端点属于(1) 或(2). 从 η 开始沿着 $\mathrm{conv}(S_{i-1})$ 的边界向两边搜索，可以找到所有类型(1) 的点和类型(2) 的两个点 v_1 和 v_2. 从链表中删去所有类型(1) 的点，把 p_i 加在 v_1 和 v_2 之间. 这些都可以在这种点的个数的线性时间内完成. 在删去 $\mathrm{conv}(S_{i-1})$ 的每一条边时，枚举 $S - S_i$ 中切割这条边的所有点. 检查它们是切割 $\overline{p_iv_1}$ 还是切割 $\overline{p_iv_2}$，进而修改它们的指针. 见图 5-2，删去 a、b、c3 点及相关的 4 条边. 把 p_i 插入 v_1 和 v_2 之间. 修改点 q 的指针，指向 $\overline{v_1p_i}$. 点 r 的指针不变.

定理 1 用上述随机增长算法计算平面上 n 个点的凸包的期望运行时间为 $O(n\mathrm{lb}n)$.

5.2 对 偶 性

定义 2 设 $a_1, a_2, \cdots, a_m$ 不全为零，称 m 维空间中的点 $p = (a_1, a_2, \cdots, a_m)$ 和超平面 $\pi: a_1x_1 + a_2x_2 + \cdots + a_mx_m + 1 = 0$ 互为对偶.

点与超平面的对偶有下述性质：设 m 维空间中的 m 个点 $p_1, p_2, \cdots, p_m$，它们的对偶分别是超平面 $\pi_1, \pi_2, \cdots, \pi_m$，则 $\pi_1, \pi_2, \cdots, \pi_m$ 的交点（假设有的话）的对偶是由 $p_1, p_2, \cdots, p_m$ 确定的超平面.

设 m 维空间中的点集 $S = \{p_1, p_2, \cdots, p_n\}$，包含 S 的最小凸集称作 S 的凸包，记作 $\mathrm{conv}(S)$. $\mathrm{conv}(S)$ 是一个 m 维凸多面体，其顶点集是 S 的子集.

不妨设原点在 $\mathrm{conv}(S)$ 的内部且不属于 S. 点 p_i 的对偶超平面为 π_i，把以 π_i 为边界包含原点的 m 维半空间记作 h_i. 令 $H = \{h_1, h_2, \cdots, h_n\}$，$H$ 中 n 个 m 维半空间的交记作 $\mathrm{inter}(H)$. $\mathrm{inter}(H)$ 也是一个 m 维凸多面体，它以 H 中的某些超平面为边界面.

conv(S) 与 inter(H) 有下述关系：conv(S) 的顶点的对偶恰好是 inter(H) 的边界面，并且 conv(S) 的两个顶点相邻当且仅当它们的对偶在 inter(H) 上相邻.

特别地，设平面上的凸包 conv(S) 的顶点按逆时针方向依次是 $p_{i_1}, p_{i_2}, \cdots, p_{i_t}$，则半平面的交 inter($H$) 的边界依次是这些点的对偶直线.

根据上述性质，求点集的凸包和求半空间的交是一对对偶问题. 很容易把一个问题的算法转化成另一个问题的算法.

5.3　半空间的交

设三维空间中 n 个半空间的集合 H，每一个半空间用坐标的一次不等式表示，容易判断任给的一点是否属于这个半空间. 当 inter(H) 非空时，它是一个凸多面体；当 inter(H) 无界时，把 ∞ 作为它的一个顶点，并且每一条半无穷长的棱都以 ∞ 为端点. 用图 $G(H)$ 表示 inter(H)，$G(H)$ 的顶点就是 inter(H) 的顶点（当 inter(H) 无界时、包括 ∞ 在内），$G(H)$ 的两个顶点之间有一条边，当且仅当在 inter(H) 上这两个顶点之间有一条棱. $G(H)$ 是一个可平面图. 不妨假设 H 中不存在 4 个半空间的边界面共点（否则其中必有一个不在 inter(H) 的边界上，可以删去）. 于是，$G(H)$ 的每一个顶点（除 ∞ 之外）的度数为 3. inter(H) 的面数不超过 n，顶点数和棱数均为 $O(n)$.

和平面凸包问题类似，半空间交的随机增长算法如下：

首先将 n 个半空间的顺序随机化，设为 $h_1, h_2, \cdots, h_n$. 记 $H_i = \{h_1, h_2, \cdots, h_i\}$，$1 \leqslant i \leqslant n$. 算法分 n 个阶段进行. 在阶段 i 把 h_i 加到 inter(H_{i-1}) 上，对 inter(H_{i-1}) 作适当的修改得到 inter(H_i). 直观上，加入 h_i，就是用 h_i 的边界面从 inter(H_{i-1}) 上割去不属于 h_i 的部分. 这就要删去 inter(H_{i-1}) 的某些顶点，另外加入一些顶点.

为了简单起见，设 inter(H_4) 是有界的，从而对每一个 $i \geqslant 4$，inter(H_i) 都是有界的. 对每一个 $h \in H - H_{i-1}$，用 $\overline{h}$ 表示 h 的补. 如果 inter(H_{i-1}) 的顶点属于 $\overline{h}$，则称这个顶点与 h 冲突. h 有一个双向指针指向一个与它冲突的顶点.

在阶段 i，加入 h_i. 若 h_i 的指针为空，则 inter(H_{i-1}) $\subseteq h_i$，从而，inter(H_i) = inter(H_{i-1})，不必做什么修改，转入下一阶段. 否则从 h_i 的指针所指向的那个顶点开始，在 $G(H_{i-1})$ 上进行搜索，搜索到达 h_i 的边界面停止，不进入 inter(H_{i-1}) $\cap h_i$. 这样搜索到 inter(H_{i-1}) 的所有应删去（不属于 h_i）的顶点和应添加的 inter(H_i) 的新顶点. 这些新顶点都在 h_i 的边界面上，是 inter(H_{i-1}) 连接在 h_i 中和 $\overline{h_i}$ 中的顶点的棱与 h_i 边界面的交点. 与此同时，还要修改每一个 $h \in H - H_i$ 的指针. 在删去 inter(H_{i-1}) 的顶点 v 时，检查 $H - H_i$ 中是否有指针指向 v 的半空间 h. 如果有的话，则需要把这个 h 的指针指向的顶点从 v 修改成与它冲突的 inter(H_i) 上的顶点 w.

类似于前面叙述的搜索过程，在 $G(H_{i-1})$ 上从 v 开始，不进入 h，第一次搜索到的 inter(H_i) 的顶点即为 w. 这就完成了阶段 i.

定理 2　用上述随机增长算法计算 n 个三维半空间的交的期望运行时间为 $O(n\lg n)$.

5.4 德劳赖三角剖分

设 $P = \{p_1, p_2, \cdots, p_n\}$ 是平面上 n 个点的集合. 平面上到 p_i 比到 P 中所有其它点都近的点的全体记作 $\mathrm{cell}(p_i)$. 每一个 $\mathrm{cell}(p_i)$ 是一个凸多边形(可能是无界的). n 个 $\mathrm{cell}(p_i)(i = 1,2,\cdots,n)$ 把平面划分成 n 个凸多边形区域,平面的这种划分称作 P 的**沃罗诺图**(Voronoi diagram),记作 $\mathrm{vor}(P)$. $\mathrm{cell}(p_i)$ 称作 p_i 的**沃罗诺胞腔**,简称**胞腔**,见图 5-3 中虚线画的图.

为简单起见,假设 P 中不存在 4 点共圆和 3 点共线的点. 沃罗诺图有下述性质:

(1) 2 个相邻胞腔 $\mathrm{cell}(p_i)$ 和 $\mathrm{cell}(p_j)$ 的边界在 p_i 和 p_j 连线的垂直平分线上.

(2) 每一个顶点的度数为 3.

(3) 若 $\mathrm{cell}(p_i)$、$\mathrm{cell}(p_j)$、$\mathrm{cell}(p_k)$ 共享一个顶点,则在由 p_i、p_j、p_k 确定的圆内不包含 P 的点.

(4) 如果 p_i 是 P 的凸包 $\mathrm{conv}(P)$ 上的顶点,则 $\mathrm{cell}(p_i)$ 无界.

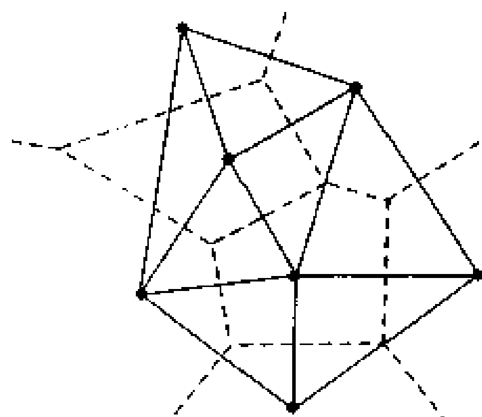


图 5-3

$\mathrm{vor}(P)$ 的对偶图称作 P 的**德劳赖三角剖分**(Delaunay triangulation),记作 $\mathrm{del}(P)$,即 $\mathrm{del}(P)$ 以 P 为顶点集,点 p_i 对应于 $\mathrm{vor}(P)$ 的面 $\mathrm{cell}(p_i)$. p_i 和 p_j 之间有一条边当且仅当 $\mathrm{cell}(p_i)$ 和 $\mathrm{cell}(p_j)$ 相邻,见图 5-3 中实线画的图. 根据 $\mathrm{vor}(P)$ 的性质(2),除最外边的面之外,$\mathrm{del}(P)$ 的面都是三角形,它把 $\mathrm{conv}(P)$ 划分成若干三角形.

构造适当的数据结构可以在时间 $O(n)$ 内由 $\mathrm{vor}(P)$ 构造出 $\mathrm{del}(P)$;反之,也可以在时间 $O(n)$ 内由 $\mathrm{del}(P)$ 构造出 $\mathrm{vor}(P)$. 因而,只要得到 $\mathrm{vor}(P)$ 和 $\mathrm{del}(P)$ 中的一个,就很容易得到另一个.

下面利用抛物变换(parabolic transformation)把计算 $\mathrm{del}(P)$ 转化成计算三维半空间的交,从而可以用随机增长算法计算 $\mathrm{del}(P)$. 考虑抛物面 $z = x^2 + y^2$,设 $p_i = (x_i, y_i, 0)$, $q_i = (x_i, y_i, x_i^2 + y_i^2)$ 是抛物面上 p_i 点正上方的点. 把抛物面在 q_i 点的切平面上面的半空间记作 h_i. 记 $H = \{h_1, h_2, \cdots, h_n\}$,则 $\mathrm{inter}(H)$ 在 xy- 平面上的垂直投影恰好是 $\mathrm{del}(P)$.

于是,有下述计算 $\mathrm{del}(P)$ 的随机增长算法:

首先用抛物变换把点集 P 变换成三维半空间集 H,用 5.3 节的随机增长算法计算 $\mathrm{inter}(H)$,最后把 $\mathrm{inter}(H)$ 垂直投影到 xy- 平面上得到 $\mathrm{del}(P)$. 算法计算 n 个点的点集的德劳赖三角剖分的期望运行时间是 $O(n\mathrm{lb}n)$.

6 随机并行算法

6.1 随机并行排序算法

6.1.1 预备知识

1. 两个简单的并行排序算法

设 n 个数 $x_1, x_2, \cdots, x_n$，不失一般性，今后总假设它们是不相同的．设 x_i 是这 n 个数中从小到大排序后的第 k 个数，称 k 是 x_i 的秩，记作 $\mathrm{rank}(x_i)$．

令

$$b_{ij} = \begin{cases} 1, & 若\ x_j < x_i, \\ 0, & 否则, \end{cases} \quad i, j = 1, 2, \cdots, n.$$

显然，

$$\mathrm{rank}(x_i) = \sum_{j=1}^{n} b_{ij} + 1, \quad i = 1, 2, \cdots, n. \tag{6-1}$$

求得每个数的秩，就等于将它们排好序．

算法 Para-Sort 1：

输入：数组 X，X 有 n 个元素．

输出：按递升顺序重新排序的 X．

步 1　for i = 1 to n do in parallel rank[i] ← 1.

步 2　for i = 1 to n do in parallel

　　for j = 1 to n do

　　if $X[i] > X[j]$ then rank[i] ← rank[i] + 1.

步 3　for i = 1 to n do in parallel X[rank[i]] ← $X[i]$.

在 CREW-PRAM 上，算法 Para-Sort1 对 n 个数排序使用 n 台处理机和时间 $O(n)$．

根据(6-1)式，用对分法可在时间 $O(\mathrm{lb}n)$ 内计算 $\mathrm{rank}(x_i)$，其代价是需要用 n^2 台处理机同时计算 n^2 个 b_{ij}．

算法 Para-Sort 2

输入：数组 X，X 有 n 个元素．

输出：按递升顺序重新排列的 X．

步 1　for i, j = 1 to n do in parallel

　　if $X[i] > X[j]$ then $b[i,j]$ ← 1 else $b[i,j]$ ← 0.

步 2　for i = 1 to n do in parallel

$$计算\ \mathrm{rank}[i] = \sum_{j=1}^{n} b[i,j] + 1.$$

步 3　for i = 1 to n do in parallel X[rank[i]] = $X[i]$.

在 CREW-PRAM 上,算法 Para-Sort 2 对 n 个数排序使用 n^2 台处理机和时间 $O(\mathrm{lb}n)$.

目前最好的并行排序算法,对 n 个数排序使用 n 台处理机和时间 $O(\mathrm{lb}n)$. 常用的奇偶排序(odd-even sorting)和双调排序(bitonic sorting)对 n 个数排序使用 n 台处理机和时间 $O(\mathrm{lb}^2 n)$.

2. 前缀和

设 n 个数 $x_1, x_2, \cdots, x_n$,称

$$s_i = \sum_{j=1}^{i} x_j, \quad i = 1, 2, \cdots, n.$$

为这 n 个数的**前缀和**(prefix sum).

不妨设 $n = 2^k$. 作一棵高度为 k 的完全正则二叉树. 开始时树叶从左到右依次等于 $x_1, x_2, \cdots, x_n$. 从树叶到树根向上进行,每个结点等于它的两个儿子之和,即以它为根的子树的树叶之和. 然后再从树根到树叶向下进行,右儿子等于它的父结点,左儿子等于它的父结点的左邻结点(如果有的话)与自身的和. 计算结束时树叶即为前缀和. 图6-1给出 $n = 8$ 的情况,其(a)是从下往上计算结束时的结果,(b)是从上往下进行计算的结果. 图中 1··2,1··4 分别表示 $x_1 + x_2, x_1 + x_2 + x_3 + x_4$,余类推.

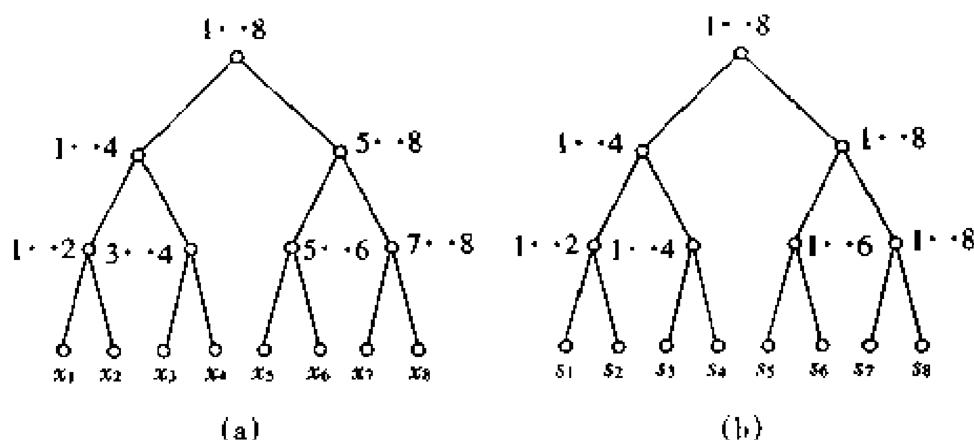


图 6-1

根据上述计算步骤,在 CREW-PRAM 上可以用 n 台处理机和时间 $O(\mathrm{lb}n)$ 计算 n 个数的前缀和.

3. 单划分元的划分

设 X 是 n 个数的集合且 n 个数不相同. 指定一个数 $y \in X$,把 $X-\{y\}$ 划分成两个子集 X_1 和 X_2, X_1 由小于 y 的数组成, X_2 由大于 y 的数组成. 把这个运算称作以 y 为划分元对 X 的划分.

设 X 以数组方式存放,划分元 $y = X[j]$, $\mathrm{rank}(y) = k$. 划分后, y 应存放于 $X[k]$. X_1 存放于 $X[1 \cdot\cdot k-1]$, X_2 存放于 $X[k+1 \cdot\cdot n]$. 为此,需要确定每个 $X[i]$ 在重新整理后的 X 中的位置 $P[i]$.

令

$$a_i = \begin{cases} 1, & \text{若 } X[i] < y, \\ 0, & \text{否则}, \end{cases} \quad i = 1,2,\cdots,n,$$

$$s_i = \sum_{t=1}^{i} a_t, \quad i = 1,2,\cdots,n,$$

则

(1) $P[j] = \text{rank}(y) = s_n + 1$.

(2) 当 $a_i = 1$ 时($X[i] < y$), $P[i] = s_i$.

(3) 当 $a_i = 0$ 且 $i \neq j$ 时($X[i] > y$), 若 $i < j$ 则 $P[i] = s_n + 1 + i - s_i$; 若 $i > j$ 则 $P[i] = s_n + i - s_i$.

在 CREW-PRAM 上能够使用 n 台处理机和时间 $O(\text{lb}n)$ 对 n 个数进行单划分元的划分.

4. 多划分元的划分

给定 m 个划分元 $y_1 < y_2 < \cdots < y_m$, 要把 n 个不同的数 $x_1, x_2, \cdots, x_n$ 插入到诸 y_i 之间, 使得 y_i 与 y_{i+1} 之间的数都插在 y_i 和 y_{i+1} 之间, $i = 0,1,\cdots,m$, 这里 $y_0 = -\infty$, $y_{m+1} = +\infty$.

使用 n 台处理机, 分两阶段进行. 第一阶段, 使用对半搜索技术确定每一个 x_j 所在段的序号 l_j, 即 $y_{l_j} < x_j < y_{l_j+1}$, 这里 $l_j \in \{0,1,\cdots,m\}$. 需 $O(\text{lb}m)$ 步. 第二阶段, 按段号从小到大排列 x_j. 这相当于给 n 个不超过 m 的非负整数排序. 为此, 构造一棵完全正则二叉树, 树的每一个结点 v 包含一个链表集合, 每个链表存放以 v 为根的子树中具相同段号的叶子. 每片叶子是一个数 x_j, 每个结点的链表集合由归并其两个儿子的链表集合得到. 计算可在 $O(\text{lb}n)$ 步内完成.

6.1.2　随机并行快速排序

将 1.1 节例 1 中算法 RandQS 并行化

算法 Para-RandQS:

输入: 数组 X, X 有 n 个元素.

输出: 按递升顺序重新排列的 X.

步 1　if $n \leqslant 1$ stop.

步 2　从数组 X 中等可能地随机取一个数 y.

步 3　以 y 作划分元, 把 X 划分成两部分. 设 $k = \text{rank}(y)$, 则 $X[k] = y$, $X[1 \cdot\cdot k-1]$ 中存放小于 y 的数, $X[k+1 \cdot\cdot n]$ 中存放大于 y 的数.

步 4　对 $X[1 \cdot\cdot k-1]$ 和 $X[k+1 \cdot\cdot n]$ 并行地递归运用算法 Para-RandQS.

步 3 使用 6.1.1 小节 3 中的算法实现.

定理 1　在 CREW-PRAM 上, 算法 Para-RandQS 对 n 个数的排序使用 n 台处理机, 在 $O(\text{lb}^2 n)$ 步内终止的概率为 $1 - O(n^{-11})$.

6.1.3　Reischuk 算法

算法 Reischuk Sort

输入: n 个数的集合 X.

输出:按递升顺序重新排列的 X.

步 1 从 X 中等可能地随机选取 $m = \lceil\sqrt{n}\rceil$ 个数 $y_1,\cdots,y_m$.

步 2 用算法 Para-Sort2 对 $y_1,y_2,\cdots,y_m$ 排序,不妨设 $y_1 < y_2 < \cdots < y_m$.

步 3 以 $y_1,y_2,\cdots,y_m$ 为划分点,用6.1.1小节4中的算法将 $X - \{y_1,y_2,\cdots,y_m\}$ 划分成 $X_0,X_1,\cdots,X_m$.

步 4 并行地处理每一个 $X_j(j = 0,1,\cdots,m)$. 若 $|X_j| > \mathrm{lb}n$ 则对 X_j 递归使用算法 Reischuk Sort;若 $|X_j| \leq \mathrm{lb}n$ 则对 X_j 使用算法 Para-Sort1.

步 5 X 按下列顺序排列:$X_0,y_1,X_1,\cdots,y_m,X_m$.

定理 2 存在常数 $\alpha > 0$,使得在 CREW-PRAM 上算法 Reischuk Sort 对 n 个数的排序使用 n 台处理机,在 $O(\mathrm{lb}n)$ 步内终止的概率不小于 $1 - \exp(-\alpha(\mathrm{lb}n)^{3/4})$.

6.2 极大独立集

设无向图 $G = (V,E)$, $|V| = n$, $|E| = m$. 如果 $I \subseteq V$ 且 I 中任意两点之间都没有边,则称 I 是一个**独立集**. G 中顶点最多的独立集称作**最大独立集**. 求最大独立集是 NP 难的. 如果 I 是独立集,并且对于每一个 $v \in V - I$, $I \cup \{v\}$ 都不是独立集,则称 I 是**极大独立集**,简记作 MIS(Maximal Independent Set). 顶点的下标字典序最小的极大独立集称作**字典序第一极大独立集**,简记作 LFMIS(the lexicographically first MIS). 容易给出求 LFMIS 的贪婪算法,其运行时间为 $O(n + m)$. 但是,已经知道 LFMIS 问题(问给定的顶点是否属于 LFMIS?)是 P 完全的,从而不存在求 LFMIS 的 NC 算法(除非 P = NC)和 RNC 算法(除非 P = RNC). RNC 是与 NC 对应的随机型语言类,它包括所有可以用随机的 NC 算法识别的语言.

下面考虑一般的求极大独立集问题,即只要求给出一个 MIS,而不必是 LFMIS. 卡普给出下述求 MIS 的贪婪算法的一般模式:

步 1 令 $I \leftarrow \varnothing$.

步 2 while $G = (V,E)$ 非空 do

(2-1) 取 G 的一个独立集 $S \subseteq V$,令 $I \leftarrow I \cup S$.

(2-2) 从 G 中删去 $S \cup \Gamma(S)$ 及其相关联的边,其中 $\Gamma(S)$ 是 S 的邻点集,即 $\Gamma(S) = \{u \in V \mid 存在\ v \in S 使得 \{u,v\} \in E\}$.

问题的关键是如何选择 S,使得每次循环能快速完成并且循环次数少.

算法 Luby MIS[①]:

输入:图 $G = (V,E)$.

输出:极大独立集 $I \subseteq V$.

步 1 令 $I \leftarrow \varnothing$, $(V',E') \leftarrow (V,E)$.

步 2 while $V' \neq \varnothing$ do

(2-1) for all $v \in V'$ do in parallel

计算 v 的度数 $d(v)$.

① 算法 Luby MIS 由鲁比(M. Luby)于 1985 年提出.

(2-2) for all $v \in V'$ do in parallel

　　if $d(v) = 0$ then $I \leftarrow I \cup \{v\}, V' \leftarrow V' - \{v\}$

　　　else 以概率 $1/2d(v)$ 标记 v.

(2-3) for all $\{u, v\} \in E'$ do in parallel

　　if u 和 v 都有标记 then 将 u、v 中度数较小者的标记抹去(当度数相等时抹去下标较小者的标记).

(2-4) 记 S 为所有带标记的顶点，

　　令 $I \leftarrow I \cup S, Y \leftarrow S \cup \Gamma(S)$.

(2-5) 令$(V', E') \leftarrow V' - Y$ 上的诱导子图.

在 EREW-PRAM 上,用 $O(n + m)$ 台处理机,每次循环可以在 $O(\text{lb}n)$ 步内完成.根据下述两个引理可以估计出循环次数.

引理 1　设在第 k 次循环开始时有 N_k 条边,经过这次循环删去 X_k 条边,则

$$E(X_k) \geqslant \frac{1}{8} N_k.$$

考虑下述模型:有 m 个位置 $1,2,\cdots,m$ 和 $m-1$ 个相互独立的随机变量 $X_2, X_3, \cdots, X_m$.每一个 X_i 服从 $\{1,\cdots,i-1\}$ 上的均匀分布.一质点开始时位于 m,到达1后试验终止.当质点位于 i 时,下一步移动到 $i - X_i (2 \leqslant i \leqslant m)$.记质点移动的次数为 T.

引理 2　设 $g: \mathbf{R}^+ \to \mathbf{R}^+$ 是非降的.如果对每一个 $2 \leqslant i \leqslant m, E(X_i) \geqslant g(i)$,则

$$E(T) \geqslant \int_1^m \frac{1}{g(x)} \mathrm{d}x,$$

其中 $\mathbf{R}^+$ 是正实数集.

把算法执行过程中每次循环开始时的边数作为质点的位置,循环次数则是质点移动次数 T.由引理1,可取 $g(x) = x/8$,得

$$E(T) \leqslant \int_1^m \frac{8}{x} \mathrm{d}x = 8\ln m.$$

于是,得到下述定理

定理 3　在 EREW-PRAM 上,算法 Luby MIS 计算 n 个顶点、m 条边的无向图的极大独立集使用 $O(n + m)$ 台处理机,期望运行时间为 $O(\text{lb}^2 n)$.

6.3　图的匹配

6.3.1　塔特矩阵与完美匹配

设无向图 $G = (V, E)$,如果 $M \subseteq E$ 并且 M 中任意两条边都不相邻,则称 M 是 G 的一个**匹配**. G 的边数最多的匹配称作 G 的**最大匹配**(maximum matching).如果 G 的每一个顶点都与匹配 M 中的一条边关联,则称 M 是**完美匹配**(perfect matching).

设 $V = \{1,2,\cdots,n\}$,对每一条边$[i,j] \in E$ 引入一个变量 $x_{ij} (i < j)$, G 的塔特

矩阵 $\boldsymbol{T}(G)=[t_{ij}]_{n\times n}$ 定义如下：

$$t_{ij}=\begin{cases}x_{ij}, & 若[i,j]\in E 且 i<j,\\ -x_{ji}, & 若[i,j]\in E 且 i>j,\\ 0, & 若[i,j]\notin E.\end{cases}$$

引理 3 图 G 有完美匹配，当且仅当 $\det(\boldsymbol{T}(G))\neq 0$.

$\det(\boldsymbol{T}(G))$ 是关于变量 x_{ij} 的 n 次多项式．根据 2.2 节中的定理 1，用指纹术容易设计出判断 $\det(\boldsymbol{T}(G))$ 是否恒等于 0 的随机算法，从而得到判断 G 是否有完美匹配的随机算法．算法如下，

算法 Perfect Matching：

输入：图 $G=(V,E)$，其中 $|V|=n$.

输出："有"或"无".

步 1　构造 G 的塔特矩阵 $\boldsymbol{T}(G)$.

步 2　for all $[i,j]\in E$ 且 $i<j$ do in parallel

　　从 $\{\pm 1,\pm 2,\cdots,\pm n\}$ 中等可能地随机取一个数赋给 x_{ij}.

步 3　计算 $\det(\boldsymbol{T}(G))$.

步 4　if $\det(\boldsymbol{T}(G))\neq 0$ then return "有" else return "无".

根据引理 6 和 2.2 节中的定理 1，当图 G 不存在完美匹配时，算法总回答"无"，当图 G 存在完美匹配时，算法回答"有"的概率不小于 1/2.

6.3.2 完美匹配的随机并行算法

算法 Perfect Matching 仅能判断是否有完美匹配，而没有求出完美匹配．当 G 有唯一的完美匹配 M 时，对每一条边 $[i,j]\in E$，删去顶点 i 和 j 及其关联的边后仍有完美匹配，当且仅当 $[i,j]\in M$．因而，对每一条边进行这样的检查就可以得到这个唯一的完美匹配 M．但是，当 G 有 2 个或 2 个以上完美匹配时，这个做法失效．为了使上述做法继续有效，需要把某一个完美匹配"孤立"出来成为"唯一的"完美匹配．

设有穷集合 $X=\{x_1,x_2,\cdots,x_m\}$，$\mathscr{F}\subseteq 2^X$，称 $(X,\mathscr{F})$ 是一个**集合系统**，X 称作全域，m 是系统的维数．给定正整数权函数 $w:X\to \mathbf{Z}^+$．对每一个 $S\in\mathscr{F}$，定义 S 的权

$$w(S)=\sum_{x\in S}w(x).$$

$\mathscr{F}$ 中权最小的集合称作最小权集合．

引理 4（孤立引理）　设 $(X,\mathscr{F})$ 是 m 维的集合系统，$w:X\to\{1,2,\cdots,2m\}$ 是一个随机权函数，对于所有的 $x\in X$，$w(x)$ 相互独立且都服从 $\{1,2,\cdots,2m\}$ 上的均匀分布，则 $\mathscr{F}$ 中存在唯一的最小权集合的概率不小于 1/2.

设赋权图 $G=(V,E)$，$V=\{1,2,\cdots,n\}$，G 的塔特矩阵 $\boldsymbol{T}(G)$．对每一条边 $[i,j]\in E(i<j)$，令 $x_{ij}=2^{w_{ij}}$，其中 $w_{ij}\in\mathbf{Z}^+$ 是边 $[i,j]$ 的权．

引理 5　如果 G 有唯一的最小权完美匹配，它的权为 W，则 $\det(\boldsymbol{T}(G))\neq 0$，并且能整除 $\det(\boldsymbol{T}(G))$ 的 2 的方幂的最大值为 2^{2W}.

引理6 设 M 是 G 的唯一最小权完美匹配，它的权为 W，则边 $[i,j] \in M$ 当且仅当 $T_{ij} \cdot 2^{w_{ij}}/2^{2W}$ 是奇数，其中 T_{ij} 是 $T(G)$ 的代数余子式.

算法 MVV[①]：

输入：图 $G=(V,E)$，G 有完美匹配，$|V|=n$，$|E|=m$.

输出：完美匹配 $M \subseteq E$.

步1 for all $[i,j] \in E(i<j)$ do in parallel

从 $\{1,2,\cdots,2m\}$ 中等可能地随机取一个数作为权 w_{ij}.

步2 构造 G 的塔特矩阵 $T(G)$，令 $x_{ij}=2^{w_{ij}}$.

步3 计算 $\det(T(G))$.

步4 计算 $W=\max\{d \mid 2^{2d} \mid \det(T(G))\}$.

步5 计算 $T(G)$ 的伴随矩阵 $\mathrm{adj}(T(G))=[T_{ji}]$.

步6 for all $[i,j] \in E(i<j)$ do in parallel

计算 $r_{ij}=T_{ij} \cdot 2^{w_{ij}}/2^{2W}$.

步7 for all $[i,j] \in E(i<j)$ do in parallel

if r_{ij} 是奇数 then 将边 $[i,j]$ 加入 M.

根据孤立引理（引理4），G 有唯一最小权完美匹配的概率不小于1/2. 当 G 有唯一最小权完美匹配时，根据引理5和引理6，步骤4计算出这个唯一最小权完美匹配的权 W，步骤7得到这个唯一最小权完美匹配 M. 算法 MVV 成功的概率不小于 1/2.

引理7 设 A 是 $n \times n$ 矩阵，其元素是 k 位整数. 存在使用 $O(n^2\mathrm{MM}(n))$ 台处理机和时间 $O(\mathrm{lb}^2 n)$ 计算 $\det(A)$ 的并行算法，其中 $\mathrm{MM}(n)$ 是计算两个 $n \times n$ 矩阵乘积所需的算术运算次数，当前最好结果是 $\mathrm{MM}(n)=O(n^{2.376})$. 存在使用 $O(n^{3.5}k)$ 台处理机和时间 $O(\mathrm{lb}^2 n)$ 计算 A^{-1} 和 $\mathrm{adj}(A)$ 的随机并行算法.

定理4 假设图 G 至少有一个完美匹配，则算法 MVV 找到 G 的一个完美匹配的概率不小于1/2，并且使用 $O(n^{3.5}m)$ 台处理机和时间 $O(\mathrm{lb}^2 n)$，其中 n 是图 G 的顶点数，m 是边数.

6.3.3 最大匹配的随机并行算法

设图 $G=(V,E)$，$|V|=n$，$|E|=m$. $M \subseteq E$ 是 G 的最大匹配，$|M|=\alpha$. 有 2α 个顶点与 M 中的边关联，称这些顶点是饱和点；还有 $n-2\alpha$ 个非饱和点. 添加 $n-2\alpha$ 个新顶点，并把它们与 V 中的每一个顶点连边，新顶点之间没有边. 把这个图记作 G'. 显然，M 可以扩大成 G' 的完美匹配. 反之，G' 的每一个完美匹配有 $n-\alpha$ 条边，其中恰有 $n-2\alpha$ 条与新顶点关联，剩下 α 条构成 G 的一个最大匹配. 因此，只要知道了 G 的最大匹配中的边数 α，就可以利用计算完美匹配的算法求得 G 的最大匹配.

对每一个 $k(0 \leqslant k \leqslant n$ 且 $n+k$ 是偶数)，在 G 上添加 k 个新顶点，并在每个新

① 算法 MVV 由穆尔莫利等3人（K. Mulmuley, U.K. Vazirani, V.V. Vaziran）于1987年提出.

顶点和 V 中的每一个顶点之间加一条边，把这个图记作 G_k. 显然，G_k 有完美匹配当且仅当 $k \geq n-2\alpha$. 于是，用对半搜索使用 $\text{lb}n$ 次算法 Perfect Matching 可以找到 α.

为了保证成功地找到 α 的概率足够大，在对半搜索过程中对每一个 k 重复运行算法 Perfect Matching $2\text{lblb}n$ 次. 只要有一次 $\det(\boldsymbol{T}(G_k)) \neq 0$，就返回"有". 只有 $2\text{lblb}n$ 次都是 $\det(\boldsymbol{T}(G)) = 0$，才返回"无". 对每一个 k，回答错误的概率不超过 $(1/2)^{2\text{lblb}n} = 1/\text{lb}^2 n$. 算法正确地得到 α 的概率不小于 $1 - \text{lb}n \cdot 1/\text{lb}^2 n = 1 - 1/\text{lb}n$.

7 去随机算法

设随机算法 A 关于实例 I 和样本点 $x \in \Omega$ 输出 $A(I,x)$. Ω 是与 I 相关联的样本空间. 对于实例 I，如果 $A(I,x)$ 是所需要的解，则称 x 是关于 I 的好的样本点. 如果有一个在 Ω 中搜索好的样本点的算法，就能除去 A 的随机性，得到一个确定算法. 这种算法称作**去随机算法**(derandomized algorithms). 最直接的去随机算法是穷举搜索 Ω. 但是，通常 $|\Omega|$ 是 $|I|$ 的指数函数. 穷举搜索 Ω 不能得到有效的去随机算法. 为了得到有效的去随机算法，必须有搜索 Ω 的有效方法.

第一个方法是**压缩样本空间法**. 设随机变量函数 $\alpha: \Omega' \to \Omega$ 满足下述条件：① $|\Omega'|$ 是 $|I|$ 的多项式界限的；② 对于 $z \in \Omega'$，$A(I, \alpha(z))$ 保留着原来所需要的统计性质，从而存在 $z^* \in \Omega'$ 使 $\alpha(z^*)$ 是好的样本点. 于是，可以通过穷举搜索 Ω' 得到有效的去随机算法. 对于 Ω' 的所有样本点，计算可以并行进行. 这个方法在本质上是并行的.

第二个方法是**条件概率法**. 搜索 Ω 的过程可以表成一棵二叉树，采用对半搜索寻找好的样本点. 在每一步，当前所在的样本子空间中有好的样本点，它被分成两部分，至少有一部分中有好的样本点. 通过计算条件概率或条件期望确定哪一部分中必有好的样本点. 问题的关键是设计出计算条件概率或条件期望的有效算法. 搜索可在 $\text{lb}|\Omega|$ 步内完成.

7.1 压缩样本空间法

(1) 给定正整数 n，取 $l = \lfloor \text{lb}n + 1 \rfloor$，则有 $n < 2l \leq 2n$. 设 $\boldsymbol{Z} = (Z_1, Z_2, \cdots, Z_l)$，其中 $Z_1, Z_2, \cdots, Z_l$ 相互独立且都服从 $p = 0.5$ 的 0-1 分布. 令

$$X_i(\boldsymbol{Z}) = \Big(\sum_{k=1}^{l} i_k \cdot Z_k\Big) \pmod 2, \quad i = 1,2,\cdots,n, \tag{7-1}$$

这里 $i_1 i_2 \cdots i_l \in \{0,1\}^l$ 是 i 的二进制表示.

定理 1 由(7-1)式给出的 n 个随机变量 $X_1(\boldsymbol{Z}), X_2(\boldsymbol{Z}), \cdots, X_n(\boldsymbol{Z})$，两两相互独立且都服从 $p = 0.5$ 的 0-1 分布.

(2) 设 n 和 k 为正整数，q 是素数且 $n \leq q \leq 2n$. $n \times q$ 矩阵 $\boldsymbol{A} = [a[i,t]]$ 的

第 i 行有 n_{ij} 个 $R_j(1 \leqslant j \leqslant r)$，这里 $\sum_{j=1}^{r} n_{ij} = q(0 \leqslant i \leqslant n-1)$. 设 $Z = (Z_0, Z_1, \cdots, Z_{k-1})$，其中 $Z_0, Z_1, \cdots, Z_{k-1}$ 相互独立且都在 $\{0,1,\cdots,q-1\}$ 上服从均匀分布. 令

$$X_i(Z) = a[i, (\sum_{t=0}^{k-1} Z_t i^t) \pmod{q}], \quad i = 0, 1, \cdots, n-1. \tag{7-2}$$

定理 2　由(7-2)式给出的随机变量 $X_0(Z), X_1(Z), \cdots, X_{n-1}(Z)$ 满足下述条件：

1° k 相互独立，即 $X_0(Z), X_1(Z), \cdots, X_{n-1}(Z)$ 中的任意 k 个都是相互独立的.

2° $P_r\{X_i(Z) = R_j\} = n_{ij}/q, 0 \leqslant i \leqslant n-1, 1 \leqslant j \leqslant r$.

(3) 顶点划分问题. 给定无向图 $G = (V, E)$，把 V 划分成两部分 C 和 $\overline{C}$，使得割集 $(C, \overline{C})$ 中的边数最多. 这是最大割集问题当所有边的权为 1 时的特殊情况.

设 $V = \{1, 2, \cdots, n\}$，随机向量 $X = (X_1, X_2, \cdots, X_n)$，其中 $X_1, X_2, \cdots, X_n$ 相互独立且都服从 $p = 0.5$ 的 0-1 分布. 令

$$C(X) = \{i \mid X_i = 1, i = 1, 2, \cdots, n\},$$

即每个顶点相互独立地以概率 1/2 随机地落入 C. 于是

$$(C(X), \overline{C}(X)) = \{[i,j] \in E \mid (X_i = 1 \wedge X_j = 0) \vee (X_i = 0 \wedge X_j = 1)\}.$$

记 $B(X) = |(C(X), \overline{C}(X))|$，则有

$$E[B(X)] = m/2. \tag{7-3}$$

这里 $m = |E|$. 这给出一个随机算法，它求得的割集大小的期望等于 $m/2$.

事实上，当 $X_1, X_2, \cdots, X_n$ 两两相互独立时，(7-3)式仍然成立. 取随机向量 $Z = (Z_1, Z_2, \cdots, Z_l)$ 满足 7.1 中的条件，由(7-1)式给出 $X(Z) = (X_1(Z), \cdots, X_n(Z))$. 由定理 1，$X_1(Z), X_2(Z), \cdots, X_n(Z)$ 两两相互独立且都服从 $p = 0.5$ 的 0-1 分布. 于是，仍然有

$$E[B(X(Z))] = m/2.$$

对每一个 $Z \in \{0,1\}^l$，求 $C(X(Z))$. 取其中最大的割集作为输出，这就得到一个去随机算法，它求得的割集不小于 $m/2$. 由于样本空间 $\{0,1\}^l$ 的大小为 $2^l \leqslant 2n$，算法是多项式时间的. 可以对所有的样本点 $Z \in \{0,1\}^l$ 并行地进行计算，需要使用 $O(n)$ 台处理机.

在这里，把 X 的大小为 2^n 的样本空间 $\{0,1\}^n$ 压缩成 Z 的大小不超过 $2n$ 的样本空间 $\{0,1\}^l$，从而实现去随机性.

7.2　条件概率法

仍以 7.1 节(3)中顶点划分问题的随机算法为例说明. 把 $X_1, X_2, \cdots, X_k$ 的值取定为 $x_1, x_2, \cdots, x_k \in \{0,1\}$ 后的 $B(X)$ 记作 $B(X \mid x_1, x_2, \cdots, x_k)$. 记

$$E_0 = E[B(X)],$$

$$E_k(x_1, x_2, \cdots, x_k) = E[B(X \mid x_1, x_2, \cdots, x_k)], \quad k = 1, 2, \cdots, n.$$

对于每一个 $k(0 \leqslant k \leqslant n-1)$，有

$$E_k(x_1,x_2,\cdots,x_k)=\frac{1}{2}\{E_{k+1}(x_1,x_2,\cdots,x_k,1)+E_{k+1}(x_1,x_2,\cdots,x_k,0)\}.$$

从而，

$$E_k(x_1,x_2,\cdots,x_k)\leqslant\max\{E_{k+1}(x_1,x_2,\cdots,x_k,1),E_{k+1}(x_1,x_2,\cdots,x_k,0)\}.$$

当 $E_{k+1}(x_1,x_2,\cdots,x_k,1)\geqslant E_{k+1}(x_1,x_2,\cdots,x_k,0)$ 时取 $x_{k+1}=1$，否则取 $x_{k+1}=0$. 这样得到一个样本点 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ 使得

$$E_0\leqslant E_1(x_1)\leqslant E_2(x_1,x_2)\leqslant\cdots\leqslant E_n(x_1,x_2,\cdots,x_n).$$

由于 $B(\boldsymbol{X}\mid x_1,x_2,\cdots,x_n)$ 是一个常数，于是有

$$B(\boldsymbol{X}\mid x_1,x_2,\cdots,x_n)=E_n(x_1,x_2,\cdots,x_n)\geqslant E_0=m/2,$$

故 $\boldsymbol{x}$ 是一个好的样本点. 剩下的问题是如何判断"$E_{k+1}(x_1,x_2,\cdots,x_k,1)\geqslant E_{k+1}(x_1,x_2,\cdots,x_k,0)$". 下述公式解决了这个问题. 记

$$\begin{aligned}\Delta_k(x_1,x_2,\cdots,x_k)&=E_{k+1}(x_1,x_2,\cdots,x_k,1)-E_{k+1}(x_1,x_2,\cdots,x_k,0)\\&=|\{i\mid i\in\Gamma(k+1),i\leqslant k\}\{x_i=0\}|-\\&\quad|\{i\mid i\in\Gamma(k+1),i\leqslant k\}\{x_i=1\}|,\end{aligned}$$

其中 $\Gamma(k+1)$ 是顶点 $k+1$ 的邻点集. 根据上述公式，在第 k 步，设已把 $\{1,2,\cdots,k\}$ 划分成两部分 C 和 $\bar{C}$. 如果 $\Gamma(k+1)$ 不在 C 中的顶点个数大于等于在 C 中的顶点个数，则把顶点 $k+1$ 放入 C 中.

去随机算法如下：

步 1　令 $C\leftarrow\{1\}$.

步 2　for $k=1,2,\cdots,n-1$ do

$$\text{if } |\Gamma(k+1)\cap C|\geqslant\frac{1}{2}|C| \text{ then } C\leftarrow C\cup\{k+1\}.$$

这是问题的贪婪近似算法.

7.3 格点近似问题

格点近似问题(the lattice approximation problem)：给定 $n\times r$ 矩阵 $\boldsymbol{C}=[c_{ij}]$ 和 r 维向量 $\boldsymbol{p}=(p_1,p_2,\cdots,p_r)$，这里每一个 $c_{ij}\in[0,1]$，每一个 p_j 是一个实数. 求一个整数向量(格点)$\boldsymbol{q}=(q_1,q_2,\cdots,q_r)$ 使得 $\boldsymbol{C}\cdot(\boldsymbol{p}-\boldsymbol{q})^{\mathrm{T}}$ 的每一个分量的绝对值都比较小，即使得

$$\Delta_i=\Big|\sum_{j=1}^{r}c_{ij}(p_j-q_j)\Big|,\quad i=1,2,\cdots,n,$$

都比较小.

不失一般性，总可以假设 $p_j\in[0,1]$，$1\leqslant j\leqslant r$. 在此假设下，总有 $q_j\in\{0,1\}$，$1\leqslant j\leqslant r$.

7.3.1 随机舍入算法

令每一个 q_j 以概率 p_j 取值 1、以概率 $1-p_j$ 取值 0($1\leqslant j\leqslant r$)，并且它们是相互独立的. 把这称作**随机舍入**(randomized rounding).

对每一个 $i(1 \leqslant i \leqslant n)$，记

$$s_i = \sum_{j=1}^{r} c_{ij}p_j, \quad \psi_i = \sum_{j=1}^{r} c_{ij}q_j,$$

则有

$$E(\psi_i) = s_i, \quad \Delta_i = |\psi_i - s_i|.$$

引理 1 设 $a_1, a_2, \cdots, a_r \in [0,1]$，随机变量 $X_1, X_2, \cdots, X_r$ 相互独立且分别服从参数为 p_j 的 0-1 分布. 记 $\Psi = \sum_{j=1}^{r} a_jX_j$，假设

$$E(\Psi) = \sum_{j=1}^{r} a_jp_j = m > 0,$$

则

1° 对任意的 $\delta > 0$，

$$P_r\{\Psi > (1+\delta)m\} < [e^{\delta}/(1+\delta)^{1+\delta}]^m.$$

2° 对任意的 $\delta \in (0,1]$，

$$P_r\{\Psi < (1-\delta)m\} < [e^{\delta}/(1+\delta)^{1+\delta}]^m.$$

记

$$B(m,\delta) = [e^{\delta}/(1+\delta)^{1+\delta}]^m, \tag{7-4}$$

对于 $x \in (0,1)$，把使得 $B(m,\delta) = x$ 的 δ 记作 $D(m,x)$，即

$$B(m, D(m,x)) = x, \tag{7-5}$$

定理 3 对任意的 $\delta \in (0,1]$，随机舍入算法给出的 0-1 近似向量 $\boldsymbol{q}$ 使得所有偏差

$$\Delta_i \leqslant s_iD(s_i, \delta/2n), \quad i = 1,2,\cdots,n.$$

的概率大于 $1-\delta$.

推论 1 存在0-1近似向量 $\boldsymbol{q}$ 使得

$$\Delta_i \leqslant s_iD(s_i, 1/2n), \quad i = 1,2,\cdots,n. \tag{7-6}$$

7.3.2 去随机算法与悲观估计法

用条件概率法可以把格点近似问题的随机舍入算法去随机得到确定算法，它求得的 $\boldsymbol{q}$ 满足(7-6)式.

称 $\boldsymbol{q}$ 不满足(7-6)式是坏事件，坏事件的概率记作 P_0. 由推论1，$P_0 < 1$. 对于每一个 $1 \leqslant j \leqslant k$，在给定 $q_1, q_2, \cdots, q_j$ 的值后坏事件的条件概率记作 $P_j(q_1, q_2, \cdots, q_j)$. 类似 7.2 节中的 $E_k(x_1, x_2, \cdots, x_k)$，由

$$P_{j-1}(q_1, q_2, \cdots, q_{j-1}) = p_jP_j(q_1, q_2, \cdots, q_{j-1}, 1) + (1-p_j)P_j(q_1, q_2, \cdots, q_{j-1}, 0),$$

可推出

$$P_{j-1}(q_1, q_2, \cdots, q_{j-1}) \geqslant \min\{P_j(q_1, q_2, \cdots, q_{j-1}, 1), P_j(q_1, q_2, \cdots, q_{j-1}, 0)\}.$$

于是，当 $P_j(q_1, q_2, \cdots, q_{j-1}, 1) \leqslant P_j(q_1, q_2, \cdots, q_{j-1}, 0)$ 时取 $q_j = 1$，否则取 $q_j = 0$. 得到

$$1 > P_0 \geqslant P_1(q_1) \geqslant P_2(q_1, q_2) \geqslant \cdots \geqslant P_r(q_1, q_2, \cdots, q_r).$$

注意到对于给定的 $q_1,q_2,\cdots,q_r$，坏事件要么发生，要么不发生，即 $P_r(q_1,q_2,\cdots,q_r)$ 要么等于1，要么等于0. 上式表明，它必等于0，从而坏事件不发生，即 $\boldsymbol{q}=(q_1,q_2,\cdots,q_r)$ 满足(7-6)式.

现在的困难是不能有效地计算条件概率 $p_j(q_1,q_2,\cdots,q_j)$. 雷阿万(P. Raghavan)于1988年提出悲观估计法(the method of pessimistic estimators)克服了这个困难.

称满足下述条件的函数族 $U_j(q_1,q_2,\cdots,q_j),j=0,1,\cdots,r$，为 $P_j(q_1,q_2,\cdots,q_j),j=0,1,\cdots,r$，的**悲观估计**：

1° $P_j(q_1,q_2,\cdots,q_j)\leqslant U_j(q_1,q_2,\cdots,q_j),0\leqslant j\leqslant r$.

2° $U_{j-1}(q_1,q_2,\cdots,q_{j-1})\geqslant\min\{U_j(q_1,q_2,\cdots,q_{j-1},1),U_j(q_1,q_2,\cdots,q_{j-1},0)\}$, $1\leqslant j\leqslant r$.

3° 可以有效地计算每一个 $U_j(q_1,q_2,\cdots,q_j),1\leqslant j\leqslant r$.

4° $U_0<1$.

在上述过程中，用 $U_j(q_1,q_2,\cdots,q_j)$ 代替 $P_j(q_1,q_2,\cdots,q_j)$，最后找到的 $\boldsymbol{q}=(q_1,q_2,\cdots,q_r)$ 有

$$P_r(q_1,q_2,\cdots,q_r)\leqslant U_r(q_1,q_2,\cdots,q_r)\leqslant U_0<1.$$

从而，q 满足(7-6)式.

对于格点近似问题的 P_j 可取悲观估计

$$U_j(q_1,q_2,\cdots,q_j)=\sum_{i=1}^{n}\Big\{e^{-t_iL_{i+}}\prod_{k=1}^{j}(q_ke^{c_{ik}t_i}+1-q_k)\cdot\prod_{k=j+1}^{r}(p_ke^{c_{ik}t_i}+1-p_k)+e^{t_iL_{i-}}\prod_{k=1}^{j}(q_ke^{-c_{ik}t_i}+1-q_k)\cdot\prod_{k=j+1}^{r}(p_ke^{-c_{ik}t_i}+1-p_k)\Big\},$$
$$j=0,1,\cdots,r,$$

其中

$$L_{i+}=s_i[1+D(s_i,1/2n)],\quad L_{i-}=s_i[1-D(s_i,1/2n)],$$
$$t_i=\ln[1+D(s_i,1/2n)].$$

7.3.3 应用举例

考虑下述**填装整数规划**(packing integer programming)：给定 $n\times r$ 的0-1矩阵 $A=[a_{ij}]$ 和正整数 k，求 $x_j\in\{0,1\},j=1,\cdots,r$，使得

$$\begin{cases}\max\ \sum_{j=1}^{r}a_{1j}x_j,\\ \text{s.t.}\ \sum_{j=1}^{r}a_{ij}x_j\leqslant k,\ i=2,\cdots,n.\end{cases}\tag{7-7}$$

首先解上述整数规划的松弛形式：求满足条件 $x_j\in[0,1],1\leqslant j\leqslant r$ 的最优解. 这是一个线性规划. 设最优解为 $x_j^*,1\leqslant j\leqslant r$，最优值为 $M^*=\sum_{j=1}^{r}a_{1j}x_j^*$. 然后考虑格点近似问题：求近似 $\boldsymbol{x}^*=(x_1^*,x_2^*,\cdots,x_r^*)$ 的0-1向量 $\boldsymbol{x}=(x_1,x_2,\cdots,$

x_r). 但是,若 x 由直接随机舍入 x^* 得到,则可能不满足约束条件(7-7). 为此引入比例因子 $v \in (0,1)$ 满足条件

$$B\left(vk, \frac{1-v}{v}\right) < \frac{1}{n},$$

这里 B 是由(7-4)式定义的函数. 令

$$x_j^s = vx_j^*, \quad 1 \leqslant j \leqslant r,$$

$$M^s = vM^*,$$

有

$$\sum_{j=1}^{r} a_{1j} x_j^s = M^s,$$

$$\sum_{j=1}^{r} a_{ij} x_j^s \leqslant vk, \quad 2 \leqslant i \leqslant n.$$

随机舍入$(x_1^s, x_2^s, \cdots, x_r^s)$ 得到0-1向量 $x = (x_1, x_2, \cdots, x_r)$, 有

$$E\left(\sum_{j=1}^{r} a_{1j} x_j\right) = M^s,$$

$$E\left(\sum_{j=1}^{r} a_{ij} x_j\right) \leqslant vk, \quad 2 \leqslant i \leqslant k.$$

于是,根据引理3, x 使得

$$\sum_{j=1}^{r} a_{1j} x_j \geqslant M^s[1 - D(M^s, 1/n)],$$

$$\sum_{j=1}^{r} a_{ij} x_j \leqslant vk[1 + D(vk, 1/n)] = k, \quad 2 \leqslant i \leqslant n$$

都成立的概率大于0,其中函数 D 由(7-5)式定义. 从而存在0-1向量 x 满足约束条件(7.7),且目标函数值不小于 $M^s[1 - D(M^s, 1/n)]$. 用7.3.2节中的去随机算法可以在多项式时间内给出填装整数规划的这样一个可行解.

7.4　极大独立集的去随机并行算法

在上一章6.2节算法Luby MIS中,给定图 $G = (V, E)$, 每一个顶点 v 对应一个随机变量 $X(v) \in \{0, 1\}$, 标记 v 当且仅当 $X(v) = 1$. 所有 $X(v)$ 相互独立且当 $d(v) > 0$ 时 $P_r\{X(v) = 1\} = p(v)$, 这里 $p(v) = 1/2d(v)$. 取素数 $n \leqslant q \leqslant 2n$, 设随机变量 ξ、η 相互独立且都在$\{0, 1, \cdots, q-1\}$上服从均匀分布. 不妨记 $V = \{0, 1, \cdots, n-1\}$. 对每一个顶点 $v \in V$, 记 $n(v) = \lfloor q/2d(v) \rfloor$, 随机变量 $X(v) = 1$ 当且仅当 $(\xi + v \cdot \eta) \pmod q < n(v)$. 由定理2, 诸 $X(v)$ 两两相互独立且 $P_r\{X(v) = 1\} = n(v)/q$. 记 $p'(v) = n(v)/q$. 当 $0 < d(v) < n/16$ 时,

$$\frac{8}{9} p(v) \leqslant p'(v) \leqslant p(v).$$

对算法 Luby MIS 作下述两点修改:

(1) 若存在 $w \in V'$ 使 $d(w) \geqslant n/16$, 则把 w 送入独立集 I. 此次循环结束.

(2) 若所有的 $v \in V'$ 都有 $d(v) < n/16$,则进行类似原算法中的运算.但是,不再以概率 $1/2d(v)$ 标记顶点 v,而是生成两个满足上述条件的随机数 ξ 和 η,标记 v 当且仅当 $(\xi + v \cdot \eta) \pmod q < n(v)$. 即,不是相互独立地生成 n 个随机数 $X(v)$,而是相互独立地生成 2 个随机数 ξ 和 η,并且当 $(\xi + v \cdot \eta) \pmod q < n(v)$ 时取 $X(v) = 1$,否则取 $X(v) = 0$. 在修改后的算法中,标记 v 的概率是 $p'(v)$.

类似 6.2 节中的引理 3,修改后的算法有下述性质.

引理 3 设在第 k 次循环开始时有 N_k 条边,经过这次循环删去 X_k 条边,则

$$E(X_k) \geqslant \frac{1}{18} N_k.$$

由 6.2 节的引理 4,修改后的算法的循环次数的期望仍为 $O(\mathrm{lb}n)$,期望运行时间为 $O(\mathrm{lb}^2 n)$.

在修改后的算法中,样本空间为 $\{0, 1, \cdots, q-1\}^2$,其大小为 $q^2 \leqslant 4n^2$. 为了去随机,每次循环对所有的样本点并行地进行计算,取其中删去边数最多的作为本次循环的计算结果.这样删去的边数不少于期望值,因而循环次数为 $O(\mathrm{lb}n)$. 把这个确定型并行算法叫做算法 Luby Derand MIS.

定理 4 在 EREW PRAM 上计算 n 个顶点、m 条边的无向图的极大独立集,算法 Luby Derand MIS 使用 $O(n^2(n+m))$ 台处理机和运行时间 $O(\mathrm{lb}^2 n)$.

把压缩样本空间法和条件概率法两种方法结合起来,在样本空间 $\{0, 1, \cdots, q-1\}^2$ 中用条件概率法搜索好的样本点,鲁比(1993) 进一步给出该问题使用 $O(n+m)$ 台处理机和运行时间为 $O(\mathrm{lb}^5 n)$ 的确定型并行算法.

参 考 文 献

1 Rajeev M, Prabhakar R. Randomized algorithms. Cambridge: Cambridge University Press, 1995.

2 Daniel P. B, Pierluigi C. Introduction to the theory of complexity. Englewood Cliffs: Prentice-Hall, 1994.

·计算机数学卷·

第12篇

算法设计与复杂性分析

编　者　卢开澄
审校者　张立昂

目　录

引　言

快速电子计算机的出现是20世纪的一件大事，它改变了这个世界的面貌．几乎处处都感到它的存在，似乎它无所不能，其实不然．什么样的问题可由计算机来解决、什么样的问题则不能，这属于可计算性理论研究的范畴．理论上可计算的问题实际是否可计算呢？则是算法与算法复杂性分析研究的内容．利用计算机解决问题的第一步是设计算法，还必须对它的时间复杂性进行分析，看看是否实际可行？因此算法设计与分析已是计算机科学的重要内容，甚至于有人说计算机科学就是研究算法的一门学科．特别是20世纪70年代以来，由此发展起来的算法复杂性理论这一分支更是一引人入胜的领域．本篇介绍若干常见的算法设计的思想及常用的算法、最后对算法复杂性理论作简单的介绍．

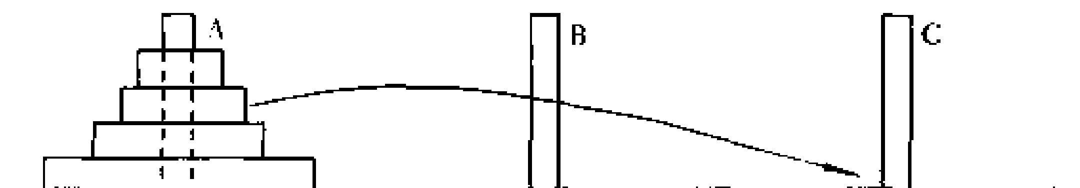
1　概　论

利用计算机求解问题，第一步是设计算法，接着是对算法所需的时空进行分析，然后编程．典型例子如汉诺（Hanoi）塔问题，这是组合数学有名的问题．设有A、B、C三根柱子，套在A柱上有 n 个盘，依大小顺序套在上面（见图1-1）．要求将它们转移到C柱上，每次只能搬动一个盘，而且不允许将大的盘子套在小的盘子的上面．问应如何操作？并估计要搬动多少盘次？

第一步，设计算法．

$n=1$ 时，很简单，直接将唯一的一个盘子转移到C柱上即可．$n=2$ 时，先将最上面的一个盘子转移到B柱上，再将A柱上剩下的一个盘子转移到C柱子上，最后将B柱上的一个盘子转移到C柱上，结束．

对于 n 个盘子可以递归地利用上述办法．先将上面 $n-1$ 个盘子转移到B柱



上，然后再将A柱上最后一个盘子转移到C柱上，最后将B柱上的 $n-1$ 个盘子转移到C柱上，结束．所以只要 $n-1$ 个盘子能做到，则 n 个盘子也能做到．

第二步，对算法的时间复杂性进行分析．

令 h_n 为 n 个盘子的汉诺塔所作的转移盘次，

$$h_n=2h_{n-1}+1,$$

因为

$$
\begin{aligned}
&h_1 = 1,\\
&h_2 = 2 + 1,\\
&h_3 = 2(2+1) + 1 = 2^2 + 2 + 1,\\
&\cdots\cdots\\
&h_{n-1} = 2^{n-2} + 2^{n-3} + \cdots + 2 + 1,\\
&h_n = 2h_{n-1} + 1 = 2^{n-1} + 2^{n-2} + \cdots + 2 + 1
\end{aligned}
$$

所以 $h_n = 2^n - 1$.

请注意，h_n 是问题的规模 n 的指数函数．n 比较大时，它实际上是不可行的．以 $n = 60$ 为例

$$2^{60} = 1.1529 \times 10^{18}.$$

每年共 $365 \times 24 \times 3600$ 秒 $= 3.1536 \times 10^7$ 秒，以每秒钟移动 1 个盘，则需时间

$$T = \left(\frac{1.1529}{3.1536}\right) \times 10^{11} \text{年} = 3.65 \times 10^{10} \text{年}.$$

又如已知序列 $x_1 < x_2 < \cdots < x_n$，给定 y，试问 y 是否在这序列中？若在，将它找出来．假定序列 $x_1, x_2, \cdots, x_n$ 是按顺序存储的．

(1) 顺序查找

步 1 $i \leftarrow 1$.

步 2 若 $x[i] > y$，则转步 5，否则转步 3.

步 3 若 $x[i] = y$，则作

【打印 i，结束】，否则作

【$i \leftarrow i + 1$，转步 4】

步 4 若 $i \leqslant n$，则转步 2，否则转步 5.

步 5 $i \leftarrow 0$，打印 i，结束．

$i = 0$ 表示 y 不在序列中，【 】是语句括号．

顺序查找最好的情况是 $y = x_1$，一次比较可找到；最坏情况为 $y > x_n$，需作 n 次比较．平均需要作 $\frac{1}{2}(n+1)$ 次比较．

(2) 二分查找．算法的思想是将 y 与 $x[\lceil \frac{n}{2} \rceil]$ 作比较，若 $x[\lceil \frac{n}{2} \rceil] = y$，则一次结束．若 $x[\lceil \frac{n}{2} \rceil] > y$，则舍弃 $x[\lceil \frac{n}{2} \rceil], x[\lceil \frac{n}{2} \rceil + 1], \cdots, x_n$ 部分，在 $x_1, x_2, \cdots, x[\lceil \frac{n}{2} \rceil - 1]$ 中去寻找．若 $x[\lceil \frac{n}{2} \rceil] < y$，则舍弃 $x_1, x_2, \cdots, x[\lceil \frac{n}{2} \rceil]$，在 $x[\lceil \frac{n}{2} \rceil + 1], x[\lceil \frac{n}{2} \rceil + 2], \cdots, x_n$ 中去寻找．一次比较删去近一半．继续以上的步骤，直到最后．

二分查找最好情况只要作一次比较，最坏情况也只要作 $\lceil \mathrm{lb}\, n \rceil$ 次比较．

二分查找算法：

步 1 $a \leftarrow 1, b \leftarrow n$.

步 2　若 $b=a$ 则作

【若 $y=x_a$ 则 $i\leftarrow a$,否则 $i\leftarrow 0$,转步 5】.

步 3　$m\leftarrow a+\lceil\frac{b-a+1}{2}\rceil-1$.

步 4　若 $y<x_m$,则作【$b\leftarrow m-1$,转步 2】,否则作

【$a\leftarrow m+1$,转步 2】

步 5　打印 i,结束．

上面讨论的查找办法可以拓广到将 y 插入到序列 $x_1,x_2,\cdots,x_n$ 中去．这时顺序存储便暴露出它的致命的弱点:平均起来将有一半的元素要改变地址．所以这时非要考虑改变序列的存储的方式不可．

2　优先策略

在算法的设计中有若干常用到的方法,优先策略是其中之一．一种办法仅用于解决一个问题,最多只能算是一种技巧,若被用来解决若干问题,则可以称之为"算法"了．按这个标准,"优先策略"是一种"算法",英文叫 Greedy method,有的译作"贪心算法".我认为还是叫"优先策略"比较贴切."优先"的含义是十分明白的,就是择优录用的意思．这仅仅是一个原则,具体的问题能不能用?如何运用这个原则?不是一件容易的事．下面通过实际问题来介绍．

2.1　求最短树的库鲁斯卡算法

设图 $G=(V,E)$是一连通无向图,$|E|=m$,$|V|=n$,已知每边的长度．所谓求最短树,即从 E 中取出 $n-1$ 条边的子集 T,使得子图(V,T)是连通的,而且 T 各边的长度之和达到最小．

库鲁斯卡(Kruskal)算法:

步 1　对属于 E 的边按边的长度进行排序,设

$$e_1\leqslant e_2\leqslant\cdots\leqslant e_m,$$

这里约定 e_i 既表示边,也用来表示边的长度．

步 2　$s\leftarrow 0,T\leftarrow\varnothing,i\leftarrow 1,t\leftarrow 0$.

步 3　若 $t=n-1$,则转步 6,否则转步 4.

步 4　若 $T\cup\{e_i\}$构成一回路,则作

【$i\leftarrow i+1$,转步 4】,否则转步 5.

步 5　$T\leftarrow T\{e_i\}$,$s\leftarrow s+e_i$,$t\leftarrow t+1$,$i\leftarrow i+1$,转步 3.

步 6　输出 T、s,结束．

其中 T 记录属于最短树的边,s 是总边长．

例1　已知图2-1.

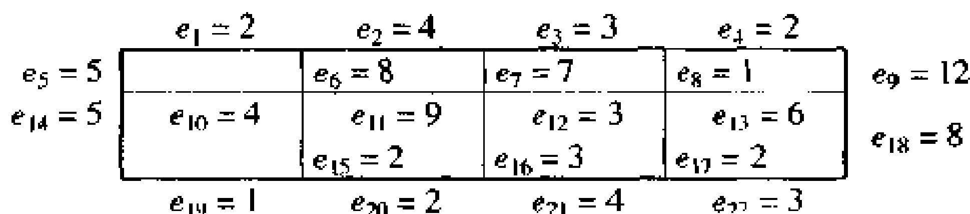


图 2-1

对边长依大小排列得

$e_8^{(1)}=1$, $e_{19}^{(2)}=1$, $e_1^{(3)}=2$, $e_4^{(4)}=2$, $e_{15}^{(5)}=2$, $e_{17}^{(6)}=2$,
$e_{20}^{(7)}=2$, $e_3^{(8)}=3$, $e_{12}^{(9)}=3$, $e_{16}^{(10)}=3$, $e_{22}^{(11)}=3$, $e_2^{(12)}=4$, $e_{10}^{(13)}=4$, $e_{21}^{(14)}=4$,
$e_5^{(15)}=5$、$e_{14}^{(16)}=5$、$e_{13}^{(17)}=6$、$e_7^{(18)}=7$, $e_6^{(19)}=8$、$e_{18}^{(20)}=8$, $e_{11}^{(21)}=9$, $e_9^{(22)}=12$.

利用库鲁斯卡算法可得最短树如图 2-2 所示．图中括弧里的数是进入 T 的顺序．

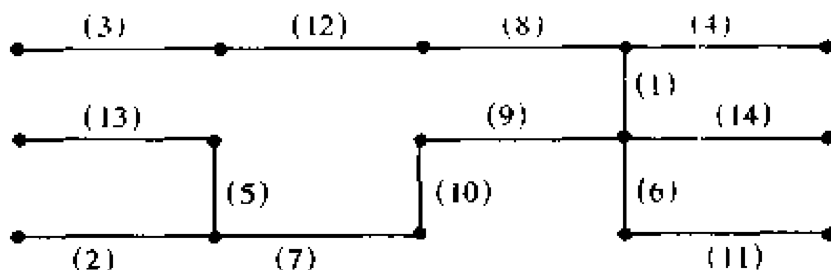


图 2-2

库鲁斯卡算法需要对所有 m 条边进行从小到大的排序，也是主要的工作量．它的时间复杂性为 $O(\mathrm{lb}m)$，这里没有估计判断是否存在回路，这个问题直观容易判断，但在计算机上则又是另一回事了．

2.2 求最短树的普林蒙算法

普林蒙(Prim)算法不同于库鲁斯卡算法，它不要求对所有的边进行排序．任取一点 v_1 作为始点，令 v_1 进入集合 S. 每一次选择一个不属于 S，但和 S 中的点距离最近的点，设为 v，即选择边$(u,v)=\min\{d_{uv}\mid u\in S,v\in \bar{S}\}$，$d_{uv}$为$(u,v)$的边长．取$(u,v)$作为树枝，并令 v 进入 S. 如此反复直到 S 包含所有 V 的点，边数达到 $n-1$ 为止．算法本身十分直观．

下面算法中用 $p[i]$记录属于 S 和 i 点最接近的点，$q[i]$记录 $i\bar{\in} S$ 且与 S 的点的最短距离．$q[i]=-1$ 表示 i 点已进入 S.

普林蒙算法：

步 1　$T\leftarrow\varnothing$，$q[1]\leftarrow-1$.

步 2　i 从 2 到 n，作

　　$\{p[i]\leftarrow1,q[i]\leftarrow d_{i1}\}$，$k\leftarrow1$.

步 3　若 $k\geq n$，则转步 8，否则作

【$\min \leftarrow \infty$，转步 4】.

步 4　j 从 2 到 n，作

若 $0 < q[j] < \min$，则作

【$\min \leftarrow q[j], h \leftarrow j$】.

步 5　$T \leftarrow T \cup ((h, p(h))), q[h] \leftarrow -1$.

步 6　j 从 2 到 n，作

若 $d_{hj} < q[j]$，则作

【$q[j] \leftarrow d_{hj}, p[j] \leftarrow h$】

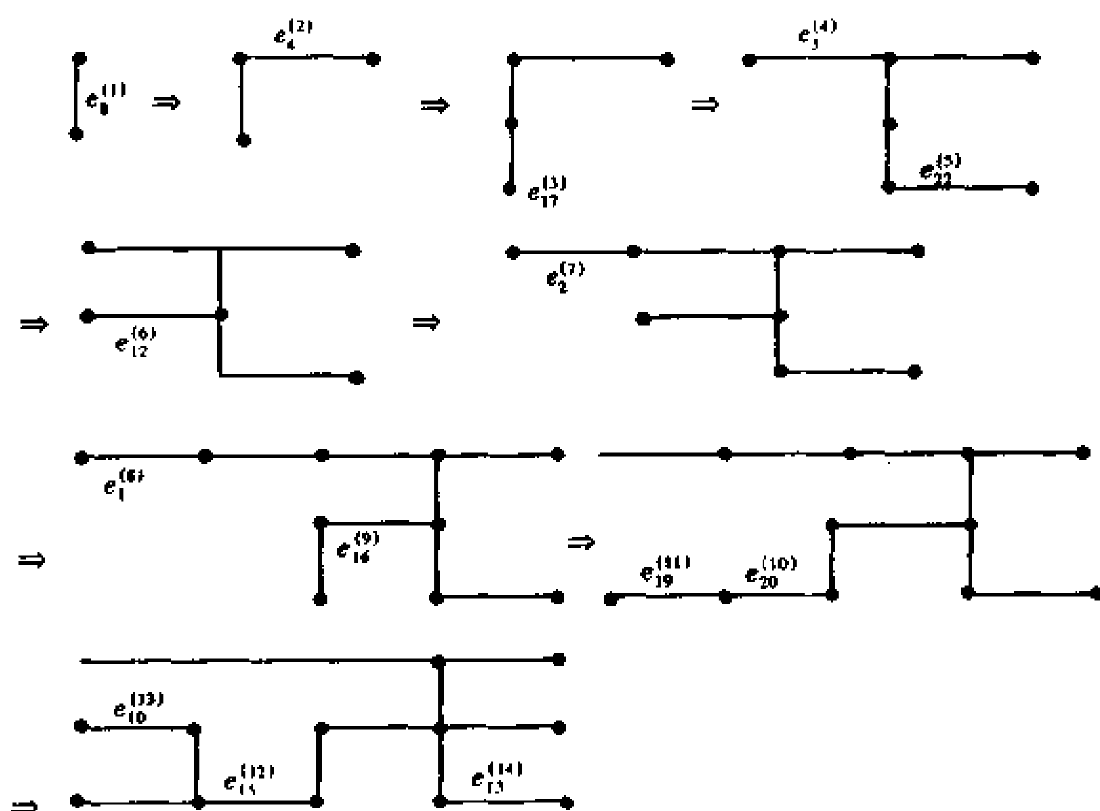
步 7　$k \leftarrow k+1$，转步 3.

步 8　输出 T，结束．

应指出的是，在步 6 中，h 点进入 S 后，应对不属于 S 中的点 j 调整 $p[j]$ 和 $q[j]$，使 $p[j]$ 记录了 S 中与 j 最接近的那一个点，相应地 $q[j]$ 记录这两点间距离．

普林蒙算法的时间复杂性为 $O(n^2)$.

例2　仍以图2-1为例，用普林蒙算法求解．



2.3　求最短路径的戴克斯徒拉算法

已知图 $G=(V,E)$，求从其中一点 v_1 到各点的最短路径的**戴克斯徒拉**(Dijkstra)算法如下：

步 1　$S \leftarrow V, T \leftarrow \varnothing, l(v_1) \leftarrow 0$,

对 $\forall v_i \in V \setminus \{v_1\}$ 作

$[l(v_i) \leftarrow \infty, p(v_i) \leftarrow \text{nil}]$

步2 若 $S=\varnothing$,则输出 l、p,结束.

步3 取 S 中 l 值最小的顶点 w, $u \leftarrow w$, $S \leftarrow S \setminus \{w\}$.

步4 对 u 的相邻顶点 v 作

若 $l(v) > l(u) + d(u,v)$,则作

$[l(v) \leftarrow l(u) + d(u,v), p(v) \leftarrow u]$,转步2.

其中 $p(v)$ 表示从初始点 v_1 到 v 路上 v 点的先行点,$p(i)=\text{nil}$ 表示 i 没有这样的点,$l(v)$ 表从 v_1 到 v 最短路径估计长度.最终结果表从 v_1 到 v 的最短路径长度.

例3 求图2-3中 O 点到各点的最短路径.

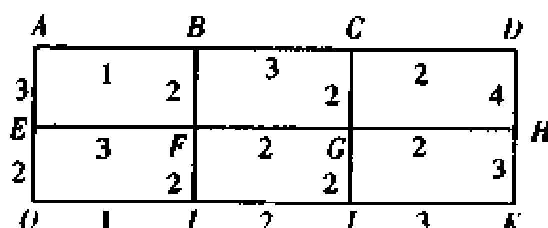
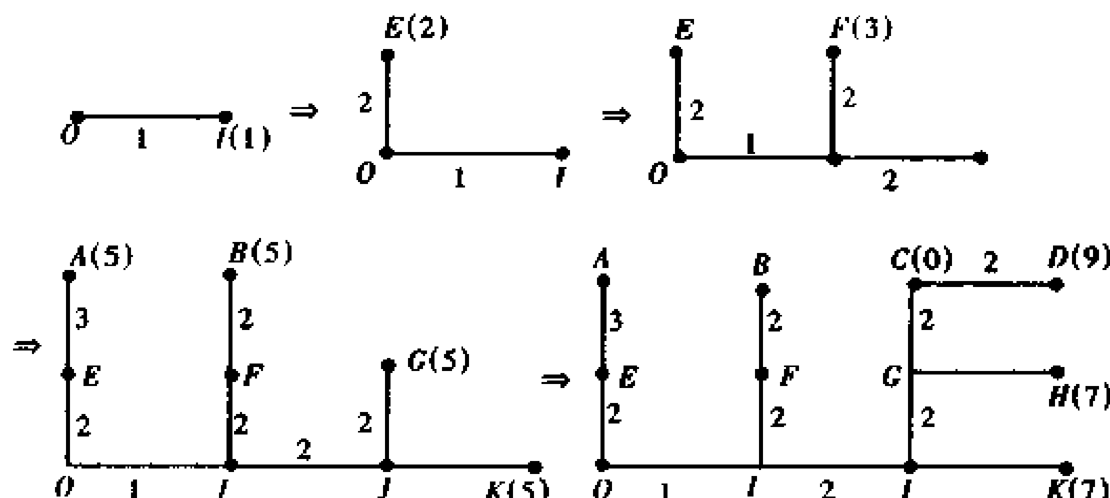


图 2-3



图中括弧里的数便是从 O 点到该点最短路径长度.

例4 有一设备,在第1,2,3,4,5年内的维修费用依次为10,15,20,30,50单位.使用1,2,3,4,5年后卖出,折旧价分别为50,40,30,20,0单位.今后5年内该设备的价格依次为100,110,115,120,130单位.问设备的更新策略应如何,使5年内费用总量最少.

问题导致求图2-4中从 v_1 点到 v_6 点的最短路径.

v_i 点代表第 i 年开始更新设备的状态.(v_i, v_j) 边上的数表示从第 i 年开始到第 j 年再更新的总费用.例如 (v_1, v_2) 边上的 $60 = 100 + 10 - 50$,即购进费用100单位,维修费用10单位,第2年更新旧机器折旧费50单位.

通过戴克斯徒拉算法可得从 v_1 到其它各点的最短路径,见图2-5.

故最佳方案是新买的设备用到第二年末,第三年初更新直到最后,总费用为

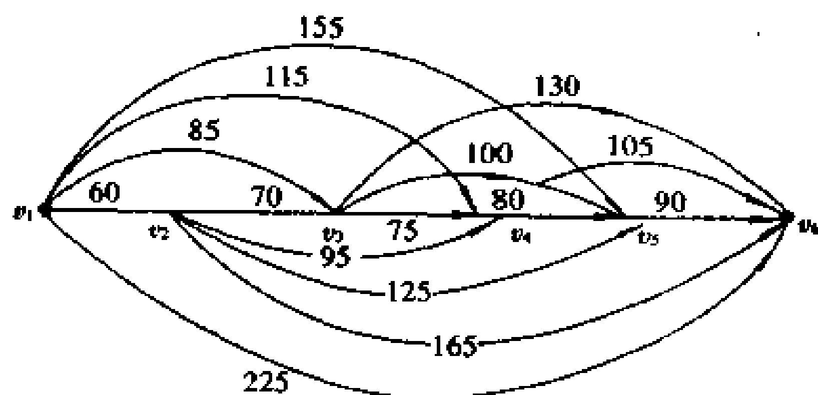


图 2-4

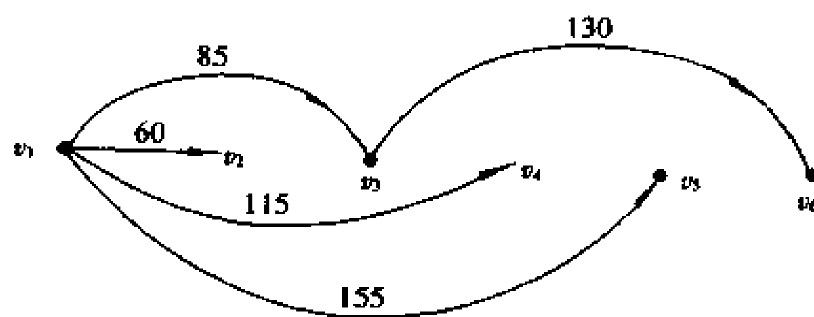


图 2-5

215(＝85＋130)单位．

2.4　磁 带 问 题

设有 n 个文件 $f_1,f_2,\cdots,f_n$，长度分别为 $l_1,l_2,\cdots,l_n$，查找的概率相等，将它们记录在一条磁带上．问文件的记录顺序应如何？

(1) 可以证明，若 $l_1\leqslant l_2\leqslant\cdots\leqslant l_n$，磁带从头到尾记录文件的顺序为

$$f_1,f_2,\cdots,f_n,$$

则文件各检索一次，磁带转动的长度

$$L=l_1+(l_1+l_2)+(l_1+l_2+l_3)+\cdots+(l_1+l_2+\cdots+l_n)$$

$$=\sum_{k=1}^{n}\sum_{j=1}^{k}l_j$$

达到最小．因

$$\begin{aligned}L=\ &l_1+\\&l_1+l_2+\\&l_1+l_2+l_3+\\&\cdots\cdots\\&l_1+l_2+l_3+\cdots+l_n\end{aligned}$$

$$= nl_1 + (n-1)l_2 + \cdots + l_n = \sum_{k=0}^{n-1}(n-k)l_{k+1}.$$

若 $k > h$(则 $l_k \geqslant l_h$),互换 f_k 和 f_h,可得更换后的磁带转动长度为

$$L_1 = nl_1 + (n-1)l_2 + \cdots + (n-h+1)l_k + \cdots + (n-k+1)l_h + \cdots + l_n,$$

$$\Delta L = L_1 - L = (n-h+1)l_k + (n-k+1)l_h - (n-h+1)l_h - (n-k+1)l_k$$
$$= (l_k - l_h)(k-h) \geqslant 0.$$

这就证明了在检索的概率相同的前提下,长度短的文件记录在前面是合适的.

(2) 若文件 $f_1, f_2, \cdots, f_n$ 的长度相等,而检索的概率 $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$,其中 p_i 是文件 f_i 检索的概率,则检索一文件磁带平均转动的长度为

$$L = \sum_{k=1}^{n} kp_k.$$

可以证明检索的概率大者记录在前面是最佳方案. 证明从略.

(3) 若文件 $f_1, f_2, \cdots, f_n$ 的长度依次为 $l_1, l_2, \cdots, l_n$,检索的概率依次为 $p_1, p_2, \cdots, p_n$,而且

$$\frac{p_1}{l_1} \geqslant \frac{p_2}{l_2} \geqslant \cdots \geqslant \frac{p_n}{l_n}.$$

若 $\frac{p_i}{l_i}$ 大的记录在前,则检索一文件磁带平均转动的长度

$$L = p_1 l_1 + p_2(l_1 + l_2) + \cdots + p_k(l_1 + l_2 + \cdots + l_k) + \cdots + p_n(l_1 + l_2 + \cdots + l_n)$$

最小.

事实上若 $\frac{p_h}{l_h} > \frac{p_{h+1}}{l_{h+1}}$,互换 f_h 和 f_{h+1},则互换后 L 的增量

$$\Delta L = p_{h+1}(l_1 + l_2 + \cdots + l_{h-1} + l_{h+1}) + p_h(l_1 + l_2 + \cdots + l_h + l_{h+1})$$
$$= p_h(l_1 + l_2 + \cdots + l_h) - p_{h+1}(l_1 + l_2 + \cdots + l_h + l_{h+1})$$
$$= p_h l_{h+1} - p_{h+1} l_h = l_h l_{h+1}\left(\frac{p_h}{l_h} - \frac{p_{h+1}}{l_{h+1}}\right) > 0,$$

即 f_h 和 f_{h+1} 互换,使 L 增加.

(4) 若文件 $f_1, f_2, \cdots, f_n$ 检索的概率相同,长度满足

$$l_1 \leqslant l_2 \leqslant \cdots \leqslant l_n,$$

将 n 个文件记录在 m 条带上,讨论它记录顺序的策略应如何,留给读者思考.

2.5　有期限的任务安排

设有 n 项任务 $J_1, J_2, \cdots, J_n$,它们完成的期限分别为 $d_1, d_2, \cdots, d_n$,在期限内完成的利润依次为 $c_1, c_2, \cdots, c_n$,假定完成每个任务的时间都是一个单位. 问应如何安排加工顺序,方使收入达到最大?

如若单纯考虑利润,则利润最高的应提在前面. 有期限的任务安排,不能仅考虑利润. 举例说明如下:例如有三项任务 J_1、J_2、J_3,完成的期限分别是 2、1、3 单位

时间,在限期内完成获利分别为 80 元、180 元、200 元.若依 c_i 的大小,加工顺序为 $J_3 \to J_2 \to J_1$,但完成 J_3 后,J_2 的期限已过,只好放弃而去完成 J_1,收入共 280 元.考虑到 J_2 的期限为 1,J_3 的期限为 3,故 J_2 可提到 J_3 前面.若加工顺序为 $J_2 \to J_1 \to J_3$,则可收入 460 元.

对有期限的任务安排,先依利润 c_i 的大小排列,设 $c_1 \geqslant c_2 \geqslant \cdots \geqslant c_n$,其中 c_i 是任务 J_i 的利润.在 $J_1 \to J_2 \to \cdots \to J_n$ 的任务安排基础上作如下的调整:安排在后面的任务可否提前,要根据情况而定;排在前面的任务利润较高,能否往后移,要看它的期限是否大于它的序号.现作非形式化的讨论如下:

假如已安排了前面 r 个任务

$$J_{h_1}, J_{h_2}, \cdots, J_{h_r},$$

它们满足条件 $d_{h_i} \geqslant i, i = 1, 2, \cdots, r$,也就是都能在期限内完成.它后面的任务 J_k,如若 $d_k \leqslant r$,能否插入到前面去?首先与 J_{h_r} 进行比较.若 $d_{h_r} = r$,J_{h_r} 正好到期,不好退让,则 J_k 只好放弃.

如若 $d_{h_r} > r, d_k = r$,则 J_k 可与 J_{h_r} 互换,即将 J_k 安插到 J_{h_r} 之前.

如若 J_k 有 $d_k < r$,则 J_k 将依次继续与 $J_{h_{r-1}}, J_{h_{r-2}}, \cdots$ 作类似的比较,看看是否有它的合适的位置,方法如上述.

算法:

步 1　按利润 c_i 大小对 J_i 进行排序得 $J_1 \to J_2 \to \cdots \to J_n$,即

$$c_1 \geqslant c_2 \geqslant \cdots \geqslant c_n.$$

步 2　$h[0] \leftarrow 0, d[0] \leftarrow 0, h[1] \leftarrow 1, k \leftarrow 1, i \leftarrow 2$.

步 3　$r \leftarrow k$.

步 4　若 $d[h[r]] \geqslant d[i]$ 则转步 5,否则转步 8.

步 5　若 $d[h[r]] \neq r$ 则作

　　　{$r \leftarrow r - 1$,转步 4},否则转步 6.

步 6　若 $d[h[r]] \geqslant d[i]$ 则转步 9,否则作 $l \leftarrow k$.

步 7　若 $l > r$ 则作

　　　{$h[l+1] \leftarrow h[l], l \leftarrow l - 1$,转步 7},否则转步 8.

步 8　$k \leftarrow k + 1, h[r+1] \leftarrow i$.

步 9　$i \leftarrow i + 1$.

步 10　若 $i \leqslant n$,则转步 3,否则输出 h,结束.

其中 $h[r]$ 是任务安排过程中第 r 位任务的下标.

2.6 哈佛曼树

一般编码,码字的码长是相同的,如 ASCII 码.但若考虑到码字出现的概率差异很大,为了提高码的信息量而压缩数据,采用一种变长度的编码技术,其中最著名的当推哈佛曼(Huffman)码,哈佛曼码在现代数据压缩技术中仍扮演十分重要的角色.

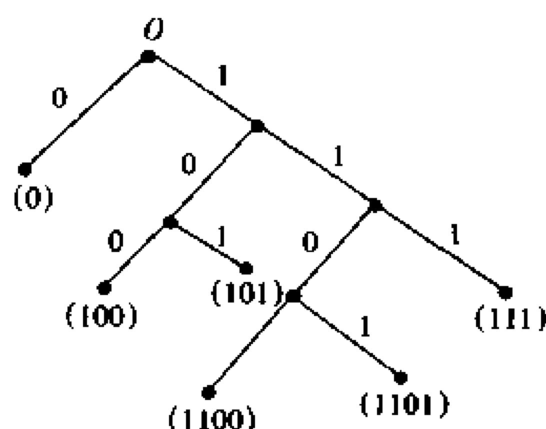


图 2-6

图 2-6 是一二元编码树．码字由 0，100，101，111，1100，1101 组成．

译码的方法举例如下：比如已知码

101 111 1100 0110

从图 2-6 的树根 O 开始向下识别，到达 101 时到达树叶节点；再从树根 O 开始，到达 111 时又到树叶；同样，到达 1100，0，1101，译码完毕．所以编码树本身也是“译码器”，称为哈佛曼树．

下面介绍如何构造哈佛曼树．

设已知 n 个字符 $a_1,a_2,\cdots,a_n$，它们出现的频率分别为 $p_1,p_2,\cdots,p_n$，

$$p_1+p_2+\cdots+p_n=1.$$

不妨假定 $p_1\leqslant p_2\leqslant\cdots\leqslant p_n$．

最佳编码是要求下列码长的数学期望

$$m=\sum_{i=1}^{n}p_il_i$$

达到最小，其中 l_i 是 a_i 的码字长度．

哈佛曼树的所有内点必有两个儿子节点，不然与最佳编码的假定相矛盾．比如图 2-7(a)的内点（*）只有一个儿子节点，则可消去“*”点得图 2-7(b)．

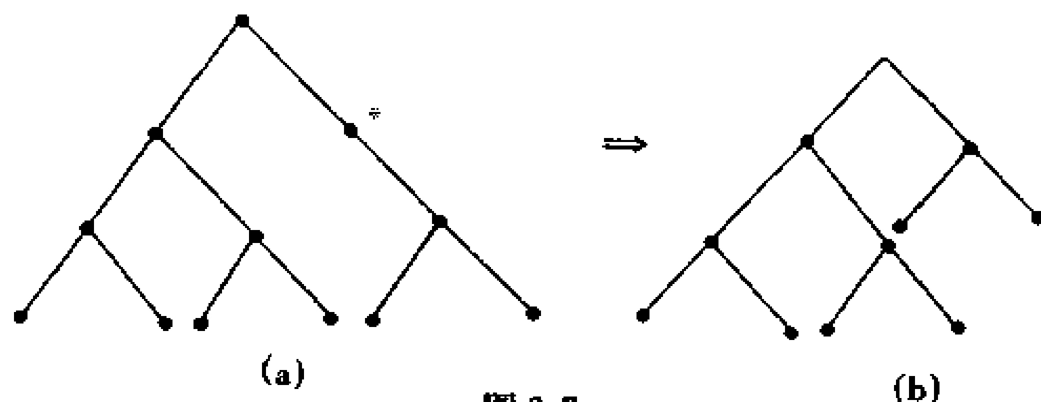


图 2-7

图 2-7(b)的码长下降，跟图 2-7(a)是最佳编码的假定相矛盾．

设 T_n^* 是对应于频率 $p_1,p_2,\cdots,p_n$ 的最佳的哈佛曼树．且 $p_1\leqslant p_2\leqslant\cdots\leqslant p_n$，则 p_1 和 p_2 是兄弟叶子节点．设 T_{n-1}^* 是对应于频率 $p_1+p_2,p_3,\cdots,p_n$ 的最佳哈佛曼树，可以证明 T_{n-1}^* 是由 T_n^* 树将 p_1+p_2 给 p_1 和 p_2 的父亲节点而得到的(图 2-8)．如此，求 T_n^* 的问题导致求 T_{n-1}^* 及求 $T_{n-2}^*,\cdots,T_3^*$．

假定 $\overline{T}_{n-1}$ 是由 T_n^* 将 p_1+p_2 给予它们的父亲节点而得到的树，证明它就是 T_{n-1}^*，即为

$$p_1+p_2,\ p_3,\ \cdots,\ p_n$$

的最佳编码树．假定 $\overline{T}_n$ 是由 T_{n-1}^* 将叶子节点 p_1+p_2 作为 p_1 和 p_2 的父亲节点而

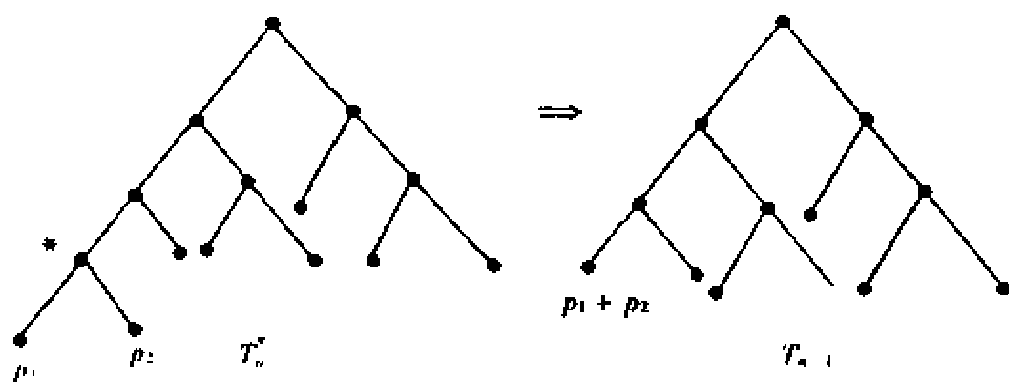


图 2-8

产生的树，如图 2-9 所示。根据假定应有

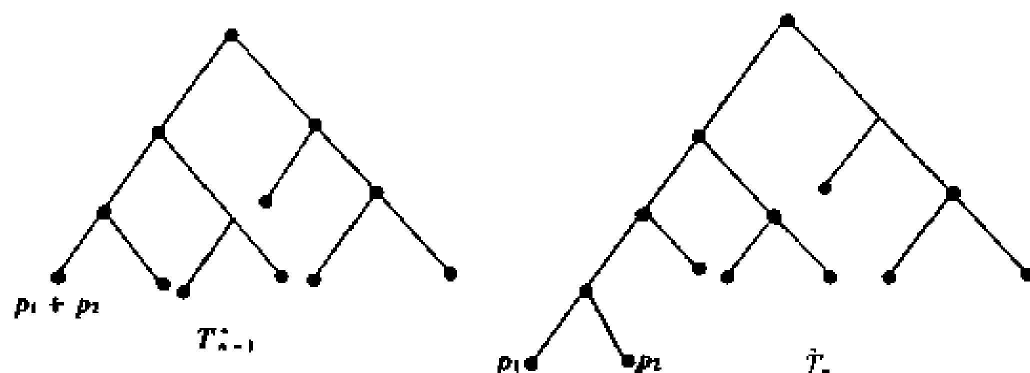


图 2-9

$$m(T_n^*)\leqslant m(\overline{T}_n),\quad m(T_{n-1}^*)\leqslant m(\overline{T}_{n-1}).$$

但

$$m(\overline{T}_{n-1})=m(T_n^*)-p_1-p_2,$$
$$m(T_{n-1}^*)=m(\overline{T}_n)-p_1-p_2,$$

根据不等式 $m(T_{n-1}^*)\leqslant m(\overline{T}_{n-1})$，所以

$$m(T_n^*)\geqslant m(\overline{T}_n),$$

导致矛盾．

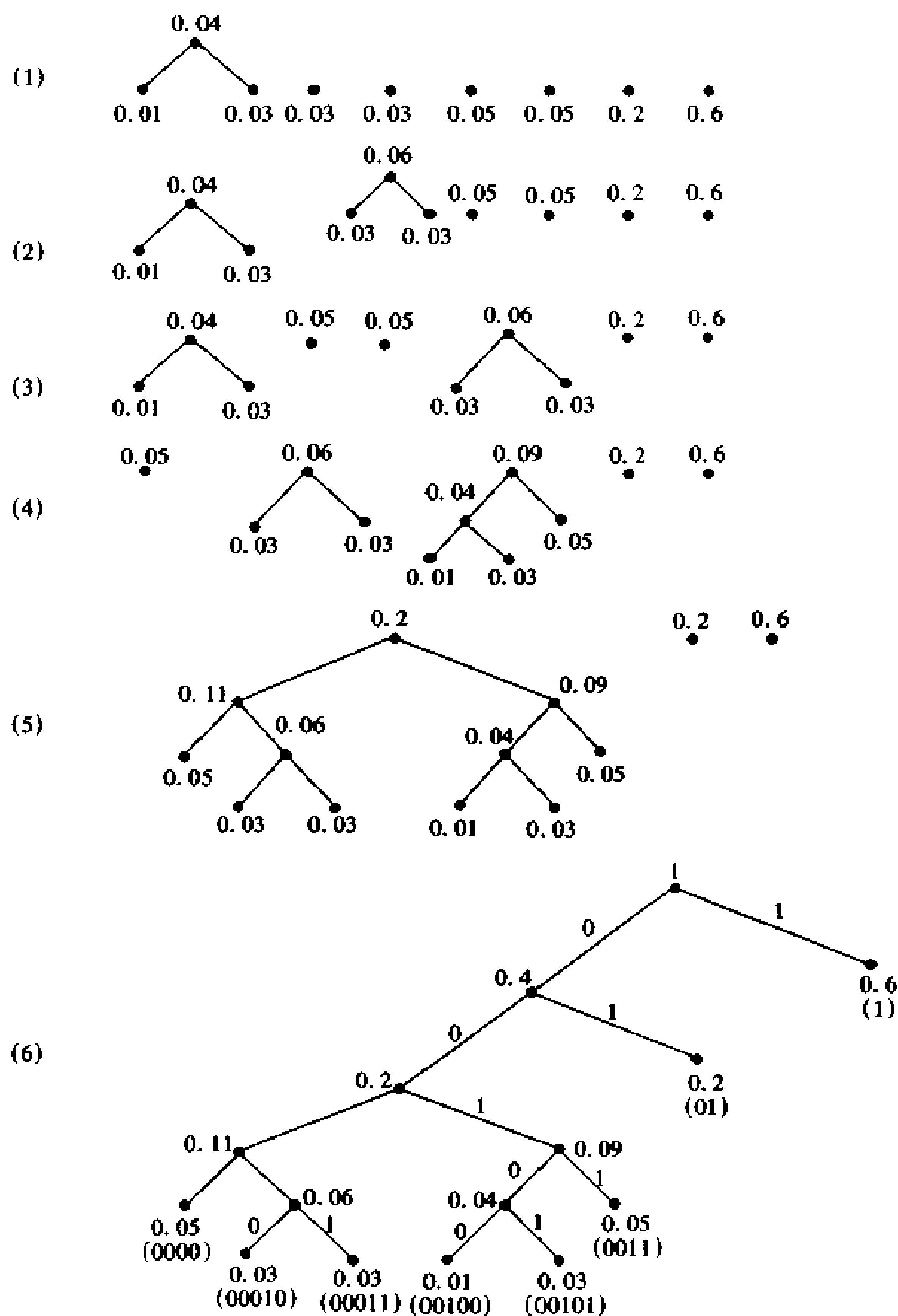
结论：求 $p_1,p_2,p_3,\cdots,p_n$ 的最佳编码树可归纳为找 $p_1+p_2,p_3,\cdots,p_n$ 的最佳编码树，并连续递归地利用这个办法．

例5 由 $a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8$ 等 8 个字符构成的文件，它们出现的概率依次为

0.6,0.2,0.05,0.05,0.03,0.03,0.03,0.01,

试构造它们的哈佛曼树．

哈佛曼树构造过程如下：


(1)
0. 04
0. 01
0. 03
0. 03
0. 03
0. 05
0. 05
0. 2
0. 6
(2)
0. 04
0. 01
0. 03
0. 06
0. 03
0. 03
0. 05
0. 05
0. 2
0. 6
(3)
0. 04
0. 01
0. 03
0. 05
0. 05
0. 06
0. 03
0. 03
0. 2
0. 6
(4)
0. 05
0. 06
0. 03
0. 03
0. 09
0. 04
0. 01
0. 03
0. 05
0. 2
0. 6
(5)
0. 2
0. 11
0. 05
0. 06
0. 03
0. 03
0. 09
0. 04
0. 01
0. 03
0. 05
0. 2
0. 6
(6)
1
0
1
0. 4
0. 6
(1)
0
1
0. 2
(01)
0. 2
1
0. 11
0. 09
0
1
0. 05
(0000)
0. 06
0
1
0. 03
(00010)
0. 03
(00011)
0. 04
0
1
0. 01
(00100)
0. 03
(00101)
0. 05
(0011)

故得编码如下：

a_1:00100	a_2:00101
a_3:00010	a_4:00011
a_5:0000	a_6:0011
a_7:01	a_8:1

3 分 治 策 略

3.1 典 型 例 子

"分治策略"源自英文 Divide and Conquer，意即分而治之的意思，即将问题分解为解若干子问题．子问题的规模和难度都要小得多．最典型的例子第 1 章中的"二分查找"，即从递增序列

$$a_1 < a_2 < \cdots < a_n$$

中寻找是否存在一元素为已知数 z. 先与 $a_{\lceil \frac{n}{2} \rceil}$比较，若 $a_{\lceil \frac{n}{2} \rceil} = z$，则一次完成．否则，若

$$z < a_{\lceil \frac{n}{2} \rceil},$$

则舍去 $a_{\lceil \frac{n}{2} \rceil+1}, a_{\lceil \frac{n}{2} \rceil+2}, \cdots, a_n$，只在序列

$$a_1, a_2, \cdots, a_{\lceil \frac{n}{2} \rceil-1}$$

中寻找．若 $z > a_{\lceil \frac{n}{2} \rceil}$，则搜索空间缩小为

$$a_{\lceil \frac{n}{2} \rceil+1}, a_{\lceil \frac{n}{2} \rceil+2}, \cdots, a_n.$$

即作一次比较，几乎将搜索的空间缩小一半．

分治策略是应用十分广的有效算法，例如多位数乘法问题．设 A 和 B 是 n 位十进制数，求 A 与 B 的乘法．按常规的办法要作 n^2 次一位数的乘法．令

$$A = a_1 \cdot 10^{\frac{n}{2}} + a_2,$$

$$B = b_1 \cdot 10^{\frac{n}{2}} + b_2,$$

假定 n 是 $2k$，a_1, a_2, b_1, b_2 都是不超过 $n/2$ 位的数，则

$$AB = a_1 b_1 10^n + (a_1 b_2 + a_2 b_1) 10^{\frac{n}{2}} + a_2 b_2. \tag{3-1}$$

令 M_k 表 2^k 位十进制数相乘时要作的一位数乘法次数，则有

$$M_k = 4M_{k-1}, \quad M_0 = 1.$$

所以
$$M_k = 4^k = n^2,$$

即按(3-1)式求 AB，须作 n^2 次一位数相乘．这就是常规的乘法法则的结果，但不是天经地义的．如采用以下的算法：

$$p = (a_1 + a_2)(b_1 + b_2),$$

$$q = a_1 b_1, \qquad r = a_2 b_2,$$

$$AB = q10^n + (p - q - r)10^{\frac{n}{2}} + r. \tag{3-2}$$

例1 设 $A = 2368, B = 3925$,令

$$a_1 = 23, a_2 = 68, b_1 = 39, b_2 = 25,$$
$$p = (23 + 68)(39 + 25) = 91 \times 64 = 5824,$$
$$q = 23 \times 39 = 897, r = 68 \times 25 = 1700,$$

则 $$AB = 897 \times 10^4 + (5800 - 897 - 1700)10^2 + 1700 = 9294400.$$

算法复杂性分析:

算法只作了 3 次 $n/2$ 位数乘法 . 令 M_k 为 2^k 位数相乘作的一位数相乘的次数,则

$$M_k = 3M_{k-1}, M_0 = 1,$$

即 $$M_k = 3^k = n^{\mathrm{lb}3}.$$

由 $n = 2^k, k = \mathrm{lb}n, 3^{\mathrm{lb}n} = n^{\mathrm{lb}3}$,说明 n 位十进制数相乘作 n^2 次一位数乘法也不是天经地义的 .

3.2 斯徒拉逊矩阵乘法

对于矩阵

$$A = \begin{bmatrix} a_{11}, a_{12} \\ a_{21}, a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

的乘积

$$C = AB = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$
$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}.$$

共有 8 次乘法 .1979 年斯徒拉逊(Strassen)提出一种只要 7 次乘法的步骤如下:

令

$$P = (a_{11} + a_{22})(b_{11} + b_{22}), \qquad Q = (a_{21} + a_{22})b_{11},$$
$$R = a_{11}(b_{12} - b_{22}), \qquad S = a_{22}(b_{21} - b_{11}),$$
$$T = (a_{11} + a_{12})b_{22}, \qquad U = (a_{21} - a_{11})(b_{11} + b_{12}),$$
$$V = (a_{12} - a_{22})(b_{21} + b_{22}),$$
$$c_{11} = P + S - T + V, \qquad c_{12} = R + T,$$
$$c_{21} = Q + S, \qquad c_{22} = P + R - Q + U.$$

算法的正确性可通过直接验证,这里从略 .

上面的矩阵乘法可推广到 A 和 B 是 $2^n \times 2^n$ 的矩阵,只要将 a_{ij} 改为 $A_{ij}, i, j = 1, 2, A_{ij}$ 为 $2^{n-1} \times 2^{n-1}$ 的矩阵 .

下面对斯徒拉逊矩阵乘法作复杂性分析 .

设 A 和 B 是 $N\times N$ 矩阵．$N=2^n$．令 p_n 为 2^n 阶矩阵作乘法运算时所作的乘法次数．由于算法将 $2^n\times 2^n$ 阶的矩阵乘法转化为 7 次 $2^{n-1}\times 2^{n-1}$阶矩阵的相乘，故

$$p_n=7p_{n-1},\quad p_1=7.$$

即

$$p_n=7^n.$$

因为 $N=2^n$，所以

$$p_n=7^{\mathrm{lb}N}=N^{\mathrm{lb}7}=N^{2.81},$$

所以斯徒拉逊矩阵乘法只作 $N^{2.81}$次乘法，而不是通常的 N^3 次乘法．

常规的矩阵乘法要作 N^3 次加法，斯徒拉逊矩阵乘法究竟要作多少次加法？分析如下：

令 a_n 表 $2^n\times 2^n$ 阶矩阵相乘时所作的加法运算次数，则

$$a_n=7a_{n-1}+18(2^{n-1})^2,\quad a_1=18.$$

这里 $18(2^{n-1})^2$ 是指附加的 18 个 $2^{n-1}\times 2^{n-1}$阶矩阵相加．令

$$G(x)=a_1+a_2x+a_3x^2+\cdots,$$
$$x:a_2=7a_1+18\cdot 2^2,$$
$$x^3:a_3=7a_2+18\cdot 2^4,$$
$$\cdots\cdots$$

所以

$$G(x)-18=7xG(x)+18\frac{4x}{1-4x},$$

$$(1-7x)G(x)=18\left(\frac{4x}{1-4x}+1\right)=\frac{18}{1-4x}.$$

令

$$G(x)=\frac{18}{(1-4x)(1-7x)}=\frac{A}{1-4x}+\frac{B}{1-7x}.$$

解得

$$A=-24,\quad B=42,$$

$$G(x)=6\sum_{k=0}^{\infty}(7^{k+1}-4^{k+1})x^k.$$

所以

$$a_n=6(7^n-4^n).$$

由 $N=2^n$，$n=\mathrm{lb}N$，$7^{\mathrm{lb}N}=N^{\mathrm{lb}7}$，故又

$$a_n=6(N^{\mathrm{lb}7}-N^2)<6N^{\mathrm{lb}7}=6N^{2.81}.$$

这说明加法的运算量也是 $O(N^{2.81})$.

最后讨论存储单元，亦即空间复杂性分析．

常规的矩阵乘法需要存放 A、B、C 三个矩阵，共 $3N^2$ 个存储单元．但斯徒拉逊乘法除 A、B 两矩阵外，还需要计算 P、Q、R、S、T、U、V．设 s_n 为 $2^n\times 2^n$ 阶矩阵乘法所需的存储单元，

$$s_n=7s_{n-1}+2(2^n)^2,\quad s_1=15.$$

类似上述可得

$$s_n=\frac{11}{3}N^{2.81}-\frac{8}{3}N^2,$$

其中 $N=2^n$，即斯徒拉逊算法的空间复杂性高于常规算法．

理论上分析时间复杂性比常规的乘法要好．实际上斯徒拉逊算法并不很实用．理想状态是 $N=2^n$．如若不然应用它程序上比较复杂．

3.3　布尔矩阵乘法

对于布尔矩阵 $\boldsymbol{A}=[a_{ij}]_{m\times n}$，$\boldsymbol{B}=[b_{jk}]_{n\times p}$．求 $\boldsymbol{A}\wedge\boldsymbol{B}=[c_{ij}]_{m\times p}$，其中

$$c_{ij}=\bigvee_{k=1}^{n}(a_{ik}\wedge b_{kj}),\quad \begin{array}{l} i=1,2,\cdots,m,\\ j=1,2,\cdots,p. \end{array}$$

布尔矩阵的逻辑乘在算法上和矩阵的乘法类似，只要将"×"代以逻辑乘"∧"，求和代以逻辑和"∨"．但布尔矩阵的每一元素不必占一单元，可用一比特表示它．

所谓布尔矩阵的乘法，有的也称它为三个苏联人的算法，先举例说明如下：

$$\boldsymbol{A}\wedge\boldsymbol{B}=\begin{bmatrix}1&1&0&1&0&0\\0&1&0&1&1&0\\1&0&0&0&0&0\\0&1&0&1&0&1\end{bmatrix}\begin{bmatrix}1&0&1&1\\1&1&0&0\\0&0&0&1\\0&1&0&0\\1&1&0&1\\1&1&1&0\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&1&0&1\\1&0&1&1\\1&1&1&0\end{bmatrix}=\boldsymbol{C}.$$

其实矩阵 $\boldsymbol{C}$ 的各行可表为 $\boldsymbol{B}$ 的行的"∨"运算：

$[1\ 1\ 1\ 1]=\boldsymbol{B}_1\vee\boldsymbol{B}_2\vee\boldsymbol{B}_4=[1\ 0\ 1\ 1]\vee[1\ 1\ 0\ 0]\vee[0\ 1\ 0\ 0]$

$[1\ 1\ 0\ 1]=\boldsymbol{B}_2\vee\boldsymbol{B}_4\vee\boldsymbol{B}_5=[1\ 1\ 0\ 0]\vee[0\ 1\ 0\ 0]\vee[1\ 1\ 0\ 1]$

$[1\ 0\ 1\ 1]=\boldsymbol{B}_1=[1\ 0\ 1\ 1]$，

$[1\ 1\ 1\ 0]=\boldsymbol{B}_2\vee\boldsymbol{B}_4\vee\boldsymbol{B}_6=[1\ 1\ 0\ 0]\vee[0\ 1\ 0\ 0]\vee[1\ 1\ 1\ 0]$

其中 $\boldsymbol{B}_i$ 是矩阵 $\boldsymbol{B}$ 的第 i 行行向量．

一般地，设

$$\boldsymbol{A}=[a_{ij}]_{m\times n}=\begin{bmatrix}\boldsymbol{A}_1\\\boldsymbol{A}_2\\\vdots\\\boldsymbol{A}_m\end{bmatrix},\quad \boldsymbol{A}_i=(a_{i1},a_{i2},\cdots,a_{in}),i=1,2,\cdots,m;$$

$$\boldsymbol{B}=[b_{ij}]_{n\times p}=\begin{bmatrix}\boldsymbol{B}_1\\\boldsymbol{B}_2\\\vdots\\\boldsymbol{B}_n\end{bmatrix},\quad \boldsymbol{B}_j=(b_{j1},b_{j2},\cdots,b_{jp}),j=1,2,\cdots,n;$$

$$\boldsymbol{C}=[c_{ij}]_{m\times p}=\begin{bmatrix}\boldsymbol{C}_1\\\boldsymbol{C}_2\\\vdots\\\boldsymbol{C}_m\end{bmatrix},\quad \boldsymbol{C}_k=(c_{k1},c_{k2},\cdots,c_{kp}),k=1,2,\cdots,m;$$

则

$$C_k = \bigvee_{j \in J_k} a_{kj} B_j, \quad J_k = \{ j \mid a_{kj} = 1 \}, k = 1, 2, \cdots, m.$$

更一般地，将矩阵 B 的 n 行分成 r 组，每组 q 个行向量：

(1) $B_1, B_2, \cdots, B_q$,

(2) $B_{q+1}, B_{q+2}, \cdots, B_{2q}$,

……

(r) $B_{(r-1)q+1}, B_{(r-1)q+2}, \cdots, B_n$.

对每组行向量取 k 个的组合，对每一组合进行"∨"运算，每组共作 $\sum_{k=1}^{q} \binom{q}{k} - q - 1 = 2^q - q - 1$ 次向量"∨"运算，其中 $2^q - q - 1$ 和 0-1 向量 $(\alpha_1, \alpha_2, \cdots, \alpha_q)$ 的个数相对应，但排除了全 0 向量和 q 个 $(\alpha_1, \alpha_2, \cdots, \alpha_q)$ 中只有 1 个元素非 0 的单位向量．共作 $r(2^q - q - 1)$ 次向量的"∨"运算．实际上是对 r 组分别作预处理．求 c_k 时只不过从 r 组中取相应的元素作"∨"运算．设 $r = n/q$，则算法共作

$$\frac{n}{q}[2^q - q - 1] + \frac{n^2}{q}$$

次向量"∨"运算．略去 m 项得

$$\frac{n}{q}2^q + \frac{n^2}{q},$$

若取 $q = \text{lb}n$ 或 $2^q = n$，算法需要作

$$\frac{2n^2}{\text{lb}n}$$

次向量的"∨"运算．

3.4　维纳格拉德算法

维纳格拉德(Winograd)算法如下．

设 $A = [a_{ij}]_{l \times m}, B = [b_{hk}]_{m \times n}$．令

$$A = [A_1 \ A_2 \cdots \ A_l]^{\mathrm{T}}, \ A_i = (a_{i1}, a_{i2}, \cdots, a_{im}), \ i = 1, 2, \cdots, l,$$

$$B = [B_1 \ B_2 \cdots \ B_n], \ B_j = (b_{1j}, b_{2j}, \cdots, b_{mj})^{\mathrm{T}}, \ j = 1, 2, \cdots, n,$$

$$AB = [c_{ij}]_{l \times n},$$

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj} = [a_{i1} \ a_{i2} \cdots \ a_{im}] \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{mj} \end{bmatrix}.$$

由

$$\begin{aligned} x_1 y_1 + x_2 y_2 + \cdots + x_n y_n = & (x_1 + y_2)(x_2 + y_1) + (x_3 + y_4)(x_4 + y_3) + \\ & \cdots + (x_{n-1} + y_n)(x_n + y_{n-1}) - (x_1 x_2 + x_3 x_4 + \cdots + x_{n-1} x_n) - \\ & (y_1 y_2 + y_3 y_4 + \cdots + y_{n-1} y_n), \end{aligned}$$

对矩阵 A 的每行和矩阵 B 的每列，预计算

$$a_{i1}a_{i2}+a_{i3}a_{i4}+\cdots+a_{i,m-1}a_{im},\quad i=1,2,\cdots,l,$$
$$b_{1j}b_{2j}+b_{3j}b_{4j}+\cdots+b_{m-1,j}b_{mj},\quad j=1,2,\cdots,n,$$

共作$\frac{1}{2}(l+n)m$ 次乘法．所以计算 AB 共需

$$\frac{m}{2}ln+\frac{1}{2}(l+n)m$$

次乘法．当 $l=m=n$ 时可知为

$$n^3/2+n^2.$$

4　动 态 规 划

4.1　典 型 问 题

如图 4-1 所示，求从 O 点到 U 点的最短路径，其中每条边上的数即为该边的长度．这在第二章已介绍过它的戴克斯德拉算法，下面引入另一种办法．

令 D_O 表示从 O 点到 U 点的最短路径长度，D_A，D_B，…分别表示从 A，B…点到 U 点的距离，则有

$$D_O=\min\{d_{OA}+D_A,d_{OB}+D_B\},$$

而　$D_A=\min\{d_{AD}+D_D,d_{AC}+D_C\},\cdots,D_Q=\min\{d_{SQ}+D_S,d_{ST}+D_T\}$.

如若计算从后面开始，则有

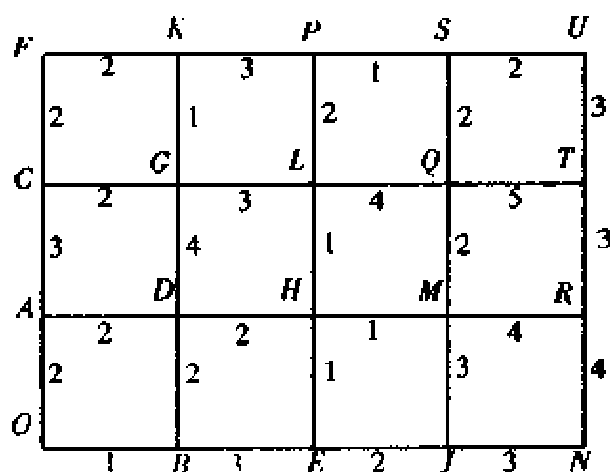


图 4-1

$$D_S=d_{SU}=2,\qquad D_T=d_{TU}=3,$$
$$D_Q=\min\{2+D_S,5+D_T\}$$
$$=\min\{2+2,5+3\}=4,$$
$$D_P=d_{PS}+D_S=1+2=3,$$
$$D_R=d_{RT}+D_T=3+3=6,$$
$$D_L=\min\{d_{LQ}+D_Q,a_{LP}+D_P\}$$
$$=\min\{4+4,2+3\}=5,$$
$$D_K=d_{KP}+D_P=3+3=6,$$
$$D_M=\min\{d_{MQ}+D_Q,d_{MR}+D_R\}$$
$$=\min\{2+4,4+6\}=6,$$
$$D_N=d_{NR}+D_R=4+6=10,$$
$$D_F=d_{FK}+D_K=2+6=8,$$
$$D_G=\min\{d_{GK}+D_K,d_{GL}+D_L\}=\min\{1+6,3+5\}=7,$$
$$D_H=\min\{d_{HL}+D_L,d_{HM}+D_M\}=\min\{1+5,3+10\}=6,$$
$$D_J=\min\{d_{JM}+D_M,d_{JN}+D_N\}=\min\{3+6,2+7\}=9,$$

$$D_C=\min\{d_{CF}+D_F,d_{CG}+D_G\}=\min\{2+8,\underline{2+7}\}=9,$$
$$D_D=\min\{d_{DG}+D_G,d_{DH}+D_H\}=\min\{4+7,\underline{2+6}\}=8,$$
$$D_E=\min\{d_{EH}+D_H,d_{EJ}+D_J\}=\min\{\underline{1+6},2+9\}=7,$$
$$D_A=\min\{d_{AC}+D_C,d_{AD}+D_D\}=\min\{3+9,\underline{2+8}\}=10,$$
$$D_B=\min\{d_{BD},d_{BE}+D_E\}=\min\{\underline{2+8},\underline{3+7}\}=10,$$
$$D_O=\min\{d_{OA}+D_A,d_{OB}+D_B\}=\min\{2+10,\underline{1+10}\}=11,$$

所以从 O 到 U 的最短距离为 11. 从 O 第一步到 B,再从 D_B 可知 B 到 D 或到 E 均可,比如 $B\to D$,再查 D_D 可得应从 $D\to H$……依次回溯可得最短路径为

$$O\to B\to D\to H\to L\to P\to S\to U.$$

也可以是

$$O\to B\to E\to H\to L\to P\to S\to U.$$

共进行 29 次加法,12 次比较.

4.2 最佳原理

所谓**最佳原理**即不论前面的状态和策略如何,今后的最优策略只取决于当前的状态.求最短路径便符合最佳原理.

符合最佳原理的问题,计算可以从后面开始.问题转为多段判决.最佳原理十分简单,但它的应用却是饶有趣味的,并非如原理本身那么直截了当.

例 1 某工厂购进1 000台机器,准备生产 P_1、P_2 两种产品.若生产 P_1 产品每年可收入 4 500 元,损坏率达 65%;若生产 P_2 产品,每台年收入为 3 500 元,损坏率为 35%.三年后这些机器均将淘汰,问应如何安排生产使三年内收入最多.计划以一年为期.

令 $P_3(n)$表示第 3 年开始时有 n 台机器的最大收益,则

$$P_3(n)=\max_{0\le x_3\le n}\{4500x_3+3500(n-x_3)\}=4500n,$$

其中 x_3 为从事 P_1 产品生产的机器数目.即 $x_3=n$,全部机器用来生产 P_1 产品.

若考虑第 2 年的生产安排,则

$$\begin{aligned}P_2(n)&=\max_{0\le x_2\le n}\{4500x_2+3500(n-x_2)+P_3(0.35x_2+0.65(n-x_2))\}\\&=\max_{0\le x_2\le n}\{4500x_2+3500(n-x_2)+4500(0.65n-0.3x_2)\}\\&=\max_{0\le x_2\le n}\{1000x_2+3500n+2925n-1350x_2\}\\&=\max_{0\le x_2\le n}\{6425n-350x_2\}=6425n,\end{aligned}$$

即 $x_2=0$,x_2 为第 2 年投入 P_1 产品生产的机器数目.$P_2(n)$为安排两年生产 n 台机器的最佳收入.同样令 $P_1(1\ 000)$为安排第一年的生产的最大收益.令 x_1 为第一年投入生产 P_1 产品的机器数目.

$$\begin{aligned}P_1(1\ 000)=\max_{0\le x_1\le 1\ 000}\{&4500x_1+3500(1000-x_1)+\\&P_2(0.35x_1+0.45(1000-x_1))\}\end{aligned}$$

$$
\begin{aligned}
&= \max_{0 \leqslant x_1 \leqslant 1000} \{4500x_1 + 3500(1000 - x_1) + \\
&\quad 6425(0.35x_1 + 450 - 0.65x_1)\} \\
&= \max_{0 \leqslant x_1 \leqslant 1000} \{3500000 + 4176250 - 927.5x_1\} \\
&= 7676250,
\end{aligned}
$$

即 $x_1 = 0$. 故前两年不安排 P_1 产品的生产,最后一年不安排 P_2 产品的生产,即前两年生产 P_2,最后一年集中生产 P_1,收入可达 7 676 250 元 .

例 2　设有资金7万元,投资给 A、B、C 三个项目 . 其利润见表 4-1.

表 4-1　　单位:万元

项目	投资额						
	1	2	3	4	5	6	7
A	0.11	0.13	0.15	0.21	0.24	0.30	0.35
B	0.12	0.16	0.21	0.23	0.25	0.29	0.34
C	0.08	0.12	0.20	0.24	0.26	0.30	0.35

投入资金 a 万元于产品 A,利润 $f_1(a)$见于表 4-2.

表 4-2

a	1	2	3	4	5	6	7
$f_1(a)$	0.11	0.13	0.15	0.21	0.24	0.30	0.35

若投入资金 a 万元于 A、B 两项目,利润

$$f_2(a) = \max_{0 \leqslant x \leqslant a} \{g_2(x) + f_1(a - x)\},$$

其中 $g_2(x)$为投资 x 单位于 B 项目的利润 . $g_2(x) + f_1(a - x)$的计算见表 4-3.

表 4-3

a	a							
	0	1	2	3	4	5	6	7
1	0.11	0.12*						
2	0.13	0.11+0.12 =0.23*	0.16					
3	0.15	0.13+0.12 =0.25	0.11+0.16 =0.27*	0.21				
4	0.21	0.15+0.12 =0.27	0.13+0.16 =0.29	0.11+0.21 =0.32*	0.23			
5	0.24	0.25+0.12 =0.33	0.15+0.16 =0.31	0.13+0.21 =0.34	0.11+0.23 =0.34*	0.25		
6	0.30	0.24+0.12 =0.36	0.21+0.16 =0.37*	0.15+0.21 =0.36	0.13+0.25 =0.36	0.11+0.25 =0.36	0.29	
7	0.35	0.30+0.12 =0.42*	0.24+0.16 =0.40	0.21+0.21 =0.42*	0.15+0.23 =0.38	0.13+0.25 =0.38	0.11+0.29 =0.40	0.34

故 $f_2(a)$ 如表4-4.

表4-4

a	1	2	3	4	5	6	7
$f_2(a)$	0.12	0.23	0.27	0.32	0.34	0.37	0.42

7万元投资于A、B、C三个项目的最大利润

$$f_3(7)=\max_{0\le x\le 7}\{g_3(x)+f_2(7-x)\},$$

$g_3(x)+f_2(7-x)$ 列表计算于表4-5.

表4-5

x	0	1	2	3	4	5	6	7
$g_3(x)$	0.00	0.08	0.12	0.20	0.24	0.26	0.30	0.35
$f_2(7-x)$	0.42	0.37	0.34	0.32	0.27	0.23	0.12	0.00
$g_3(x)+f_2(7-x)$	0.42	0.45	0.46	0.52*	0.51	0.49	0.42	0.35

所以 $f_3(7)=0.52$.

$x=3$,故投资3万元于项目C．再从表3-4回溯可知投资A和B两项目共4万元．其中B投资3万元,A项目1万元．

4.3 流动推销员问题

已知图 $G=(V,E)$ 及其距离矩阵

$$\boldsymbol{D}=[d_{ij}]_{n\times n},$$

其中 $V=\{v_1,v_2,\cdots,v_n\}$, $|V|=n$. d_{ij} 为边 (v_i,v_j) 的距离, $i,j=1,2,\cdots,n$.

从 v_1 出发遍历 $v_2,v_3,\cdots,v_n$ 一次且仅一次,最后返回 v_1,求最短的回路,这就是**流动推销员问题**．流动推销员问题是组合数学著名的难题．

下面介绍用最佳原理将问题转化为多段判决．

令 $T(v_i|V_1)$ 为从 v_i 出发遍历 V_1 中所有点一次,最后返回 v_1 的最短路径长度, V_1 是 V 的子集．

$$T(v_i|V_1)=\min_{v_j\in V_1}\{d_{ij}+T(v_j|V_1\setminus\{v_j\})\}.$$

例3 设

$$\boldsymbol{D}=\begin{matrix} v_1\\ v_2\\ v_3\\ v_4\end{matrix}\underset{\begin{matrix}v_1&v_2&v_3&v_4\end{matrix}}{\begin{bmatrix}0&8&5&6\\6&0&8&5\\7&9&0&5\\9&7&8&0\end{bmatrix}},$$

$$T(v_1|\{v_2,v_3,v_4\})=\min\{d_{12}+T(v_2|\{v_3,v_4\}),d_{13}+T(v_3|\{v_2,v_4\}),\\ d_{14}+T(v_4|\{v_2,v_3\})\}.$$

流动推销员问题符合最佳原理，现求解如下：

$$T(v_2|\varnothing) = d_{21} = 6,\quad T(v_3|\varnothing) = d_{31} = 7,$$
$$T(v_4|\varnothing) = d_{41} = 9,$$
$$T(v_2|\{v_3\}) = d_{23} + T(v_3|\varnothing) = 8 + 7 = 15,$$
$$T(v_2|\{v_4\}) = d_{24} + T(v_4|\varnothing) = 5 + 9 = 14,$$
$$T(v_3|\{v_2\}) = d_{32} + T(v_2|\varnothing) = 9 + 6 = 15,$$
$$T(v_3|\{v_4\}) = d_{34} + T(v_4|\varnothing) = 5 + 9 = 14,$$
$$T(v_4|\{v_2\}) = d_{42} + T(v_2|\varnothing) = 7 + 6 = 13,$$
$$T(v_4|\{v_3\}) = d_{43} + T(v_3|\varnothing) = 8 + 7 = 15,$$
$$T(v_2|\{v_3,v_4\}) = \min\{d_{23} + T(v_3|\{v_4\}), d_{24} + T(v_4|\{v_3\})\}$$
$$= \min\{8 + 14, \underline{5 + 15}\} = 20,$$
$$T(v_3|\{v_2,v_4\}) = \min\{d_{32} + T(v_2|\{v_4\}), d_{34} + T(v_4|\{v_2\})\}$$
$$= \min\{9 + 14, \underline{5 + 13}\} = 18,$$
$$T(v_4|\{v_2,v_3\}) = \min\{d_{42} + T(v_2|\{v_3\}), d_{43} + T(v_3|\{v_2\})\}$$
$$= \min\{\underline{7 + 15}, 8 + 15\} = 22,$$
$$T(v_1|\{v_2,v_3,v_4\}) = \min\{d_{12} + T(v_2|\{v_3,v_4\}, d_{13} + T(v_3|\{v_2,v_4\}),$$
$$d_{14} + T(v_4|\{v_2,v_3\})$$
$$= \min\{8 + 20, \underline{5 + 18}, 6 + 22\} = 23.$$

故最佳路径是 $v_1 \to v_3 \to v_4 \to v_2 \to v_1$，总长为 23. 和前面一样，回溯 $T(v_3|\{v_2, v_4\})$ 可知 $v_3 \to v_4$.

动态给流动推销员问题提出一种解法，可是它不是一种“好算法”，也就是说它的时间复杂性和空间复杂性都是指数型的，特别是时间复杂性，以 $T(v_1|V\setminus\{v_1\})$ 为例，需要作 $n-1$ 次加法和 $n-2$ 次比较．$T(v_j|V\setminus\{v_1,v_j\})$ 要作 $n-2$ 次加法和 $n-3$ 次比较．依此类推，故加法总数为

$$A = (n-1) + (n-1)\{(n-2)C(n-2,n-2) +$$
$$(n-3)C(n-2,n-3) + \cdots + C(n-2,1)\}.$$

但

$$(n-2)C(n-2,n-2) + (n-3)C(n-2,n-3) + \cdots + C(n-1,1)$$
$$= (n-2)2^{n-3}.$$

所以
$$A = (n-1)[1 + (n-2)2^{n-3}].$$

4.4 矩阵链乘问题

先举一例，设 $A_1 = [a_{ij}^{(1)}]_{10\times 100}$，$A_2 = [a_{ij}^{(2)}]_{100\times 5}$，$A_3 = [a_{ij}^{(3)}]_{5\times 50}$，求 $A_1A_2A_3$. 由于

$$(A_1A_2)A_3 = A_1(A_2A_3) = A_1A_2A_3,$$

结果虽一样，然而所作乘法次数却差别很大．

$(A_1A_2)A_3$ 所作的乘法数为

$$10\times 100\times 5+10\times 5\times 50=7500,$$

但 $A_1(A_2A_3)$所作的乘法数为

$$100\times 5\times 50+10\times 100\times 50=75000.$$

对于 n 个矩阵的链乘

$$A_1A_2\cdots A_n,$$

其中 $A_k=[a_{ij}^{(k)}]_{r_k\times r_{k+1}}$, $k=1,2,\cdots,n$, $r_1,r_2,\cdots,r_k,r_{k+1}$都是正整数,如何确定求积的顺序使乘法的次数最少?

令 m_{ij}表求$A_iA_{i+1}\cdots A_j$的最小乘法数,

$$m_{ij}=\min_{i\leqslant k<j}\{m_{ik}+m_{k+1,j}+r_ir_{k+1}r_{j+1}\},$$
$$m_{ii}=0,\quad i,j=1,2,\cdots,n.$$

以 $n=4, r_1=30, r_2=35, r_3=15, r_4=5, r_5=10$ 为例,

$m_{12}=30\times 35\times 15=15750$,

$m_{23}=35\times 15\times 5=2625$,

$m_{34}=15\times 5\times 10=750$,

$m_{13}=\min\{m_{12}+30\times 15\times 5, m_{23}+30\times 35\times 5\}$

$\quad=\min\{15750+2250, \underline{2625+5250}\}=7875$,

$m_{24}=\min\{m_{23}+35\times 5\times 10, m_{34}+35\times 15\times 10\}$

$\quad=\min\{\underline{2625+1750}, 750+5250\}=4375$,

$m_{14}=\min\{m_{12}+m_{34}+30\times 15\times 10, m_{13}+30\times 5\times 10, m_{24}+30\times 35\times 10\}$

$\quad=\min\{15750+4500, \underline{7875+30\times 5\times 10}, 4375+10500\}=9375$,

所以最佳的乘积步骤为

$$(A_1A_2A_3)A_4.$$

再从回溯 m_{13}可知

$$(A_1(A_2A_3))A_4$$

是最佳方案.乘法次数为9375.

4.5 最长公共子序列

已知两个序列

$$A=\{a_1,a_2,\cdots,a_m\},\quad B=\{b_1,b_2,\cdots,b_n\},$$

若存在序列 $C=\{c_1,c_2,\cdots,c_l\}$使得存在单调增整数序列

$$i_1<i_2<\cdots<i_l,\quad j_1<j_2<\cdots<j_l,$$

使得
$$c_k=a_{i_k}=b_{j_k},\quad k=1,2,\cdots,l,$$

则称序列 C 为 A 和 B 的公共子序列.使 l 达到最大的公共子序列,称之为**最长公共子序列**.

例4 设 $A=\{a,b,c,b,d,a,c,b\}$, $B=\{b,d,c,a,b\}$,则 $C_1=\{b,a,b\}$是公

共子序列，$C_2=\{b,d,a,b\}$，$C_3=\{b,c,a,b\}$，$C_4=\{b,d,c,b\}$也是公共子序列，而且是最长公共子序列．

A 和 B 的最长公共子序列用

$$\mathrm{LCS}\{A,B\}$$

表示．对于 $A=\{a_1,a_2,\cdots,a_m\}$，令 $A_0=\varnothing$，$A_1=\{a_1\}$，$A_2=\{a_1,a_2\}$，$\cdots$，$A_m=\{a_1,a_2,\cdots,a_m\}=A$.

同样对于 $B=\{b_1,b_2,\cdots,b_n\}$，令 $B_0=\varnothing$，$B_1=\{b_1\}$，$\cdots$，$B_n=\{b_1,b_2,\cdots,b_n\}=B$，则

$$\mathrm{LCS}\{A_i,B_j\}=\begin{cases}\varnothing, & i,j=0,\\ \mathrm{LCS}\{A_{i-1},B_{j-1}\}\cup\{a\}, & \text{若 } a_i=b_j=a,\ i,j>0,\\ \max\{\mathrm{LCS}\{A_{i-1},B_j\},\mathrm{LCS}\{A_i,B_{j-1}\}\}, & \text{其它}.\end{cases}$$

令 $l(i,j)$表示 $\mathrm{LCS}\{A_i,B_j\}$的长度，

$$l(i,j)=\begin{cases}0, & i,j=0,\\ l(i-1,j-1)+1, & \text{若 } a_i=b_j=a,\ i,j>0,\\ \max\{l(i-1,j),l(i,j-1)\}, & \text{其它}.\end{cases}$$

求 $l(i,j)$的动态规划算法如下：

步1　$l(i,0)\leftarrow 0$，$i=1,2,\cdots,m$，
　　$l(0,j)\leftarrow 0$，$j=1,2,\cdots,n$，
　　$i\leftarrow 1$.

步2　若 $i\leqslant m$　则作
　　[$j\leftarrow 1$，转步3]，否则转步7.

步3　若 $j\leqslant n$ 则转步4，否则作
　　[$i\leftarrow i+1$，转步2].

步4　若 $a_i=b_j$，则作
　　[$l(i,j)\leftarrow l(i-1,j-1)+1$，转步3]，
　　否则转步5.

步5　若 $l(i-1,j)\geqslant l(i,j-1)$则作
　　[$l(i,j)\leftarrow l(i-1,j)$，$j\leftarrow j+1$，转步3]，
　　否则转步6.

步6　$l(i,j)\leftarrow l(i,j-1)$，$j\leftarrow j+1$，转步3.

步7　打印 $l(m,n)$，结束．

求 $l(m,n)$的算法与求 $\mathrm{LCS}(A,B)$是一致的，稍加修改即可用来计算 $\mathrm{LCS}(A,B)$．此留给读者自己来做．

例5　设 $A=\{1,2,3,2,4,1,2\}$，$B=\{2,4,3,1,2,1\}$，求最大公共子序列长度．

列表计算于表4-6.

$i=0$ 行及 $j=0$ 列全部为0，计算先自 $i=1$ 行开始自上而下，再自左而右进行．比如 $i=3,j=2$，由于 $a_i=b_j=3$．故 $l(3,2)=l(2,1)+1=2$，又 $i=3,j=3$，由于 $a_i\neq b_j$，故

$$l(3,3)=\max\{l(3,2),l(2,3)\}=\max\{2,1\}=2.$$

表 4-6

	i	j						
		0	1	2	3	4	5	6
$A\downarrow$	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	1
2	2	0	1	1	1	1	2	2
3	3	0	1	2	2	2	2	2
2	4	0	1	2	2	2	3	3
4	5	0	1	2	3	2	3	3
1	6	0	1	2	3	3	3	4
2	7	0	1	2	3	3	4	4
		$B\rightarrow$	2	3	4	1	2	1

4.6 应用举例

例6 n 块银币 $C_1,C_2,\cdots,C_n$，其中有一块不合格，它比正常的重，现用天平将它找出来．要求不用砝码，求在最坏情况下用天平次数最少的策略．

若从 n 个银币中取 $2k$ 个，天平的两边各 k 个．若天平两边平衡，说明所取的 $2k$ 个都是合格的，不合格的在余下 $n-2k$ 中；若天平两端不平衡，则不合格的银币在重的一端．

令 b_n 表示 n 个银币在最坏情况下用天平最少的次数，有

$$b_n = 1 + \min_{1\leqslant k\leqslant\lceil\frac{n}{2}\rceil}\{\max(b_{n-2k},b_k)\}.$$

于是有

$$b_0=0,\quad b_1=0,\quad b_2=1,\quad b_3=1,$$

$$b_4 = 1+\min\{\max(b_2,b_1),\max(b_0,b_2)\}$$
$$= 1+\min\{b_2,b_2\} = 1+1=2,$$

即 $n=4$ 时策略有图 4-2 中所示两种

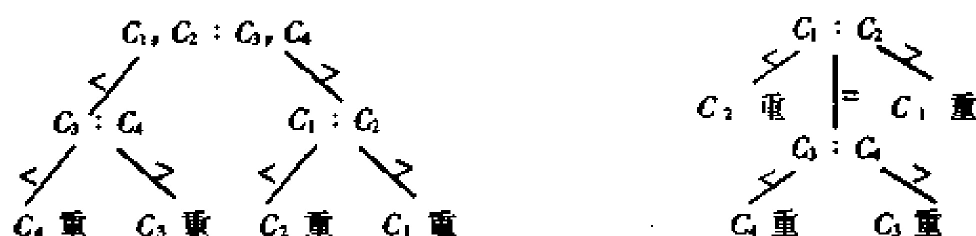


图 4-2

$n=5$ 时，

$$b_5 = 1+\min\{\max\{b_3, b_1\}, \max\{b_1, b_2\}\}$$
$$= 1+\min\{b_3, b_2\} = 2,$$

见图 4-3，

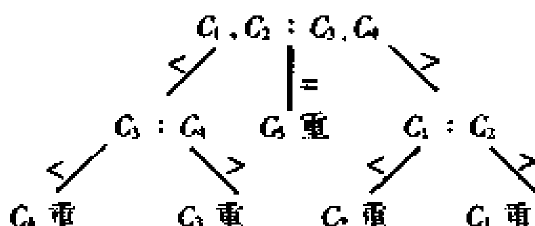


图 4-3

$n=6$ 时，

$$b_6 = 1+\min\{\max\{b_4, b_1\}, \max\{b_2, b_2\}, \max\{b_3, b_0\}\}$$
$$= 1+\min\{b_4, b_2, b_3\} = 2,$$

见图 4-4.

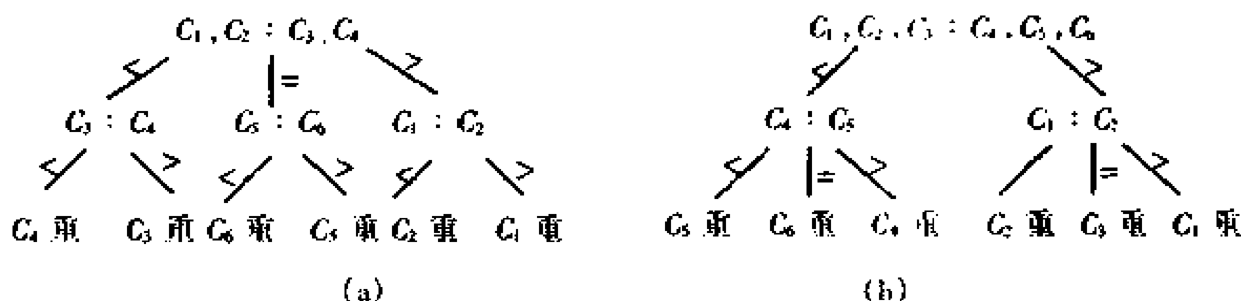


图 4-4

例 7　对于重要的装置往往会提出这样的问题：如何保证设备的可靠性达到足够的程度．一个系统可以看作是由输入经过 n 级到达输出，如图 4-5.

图 4-5

为了提高可靠性，每一级可用多个元器件并联，如图 4-6.

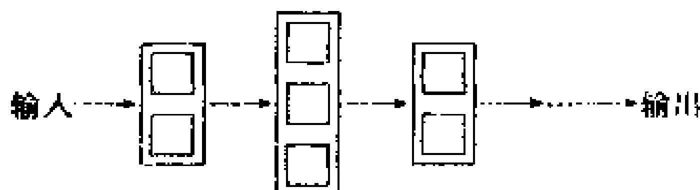


图 4-6

理论上可通过增加元器件使每一级的可靠性达到任意高的程度．实际上这是办不到的．因为重量、代价、体积等条件限制，使得元件数目不能无限制地增加．

已知三级系统分别用 D_1、D_2、D_3 三种元件，其价格和可靠性如下：

元件	D_1	D_2	D_3
单价	20	15	30
可靠性	0.5	0.8	0.9

若限制条件是代价不超过 105 元,问应如何设计使可靠性达到最大.

令 $q_1(u)$表示由 D_1 组成一级,代价限制在 u 条件下系统的可靠性,

$$q_1(u) = \max_{1\leq x_3\leq\lceil\frac{u}{20}\rceil}\{1-(0.5)^{x_3}\};$$

$q_2(u)$表示由 D_1 和 D_2 组成两级,代价为 u 条件下系统的可靠性,

$$q_2(u) = \max_{1\leq x_2\leq\lceil\frac{u}{15}\rceil}\{[1-(0.2)^{x_2}]q_1(u-15x_2)\};$$

问题是求 $q_3(105) = \max\limits_{1\leq x_1\leq\lceil\frac{105}{30}\rceil}\{[1-(0.1)^{x_1}]q_2(u-30x_1)\}$.

$$q_1(60) = \max\{0.5,0.75,0.875\} = 0.875,$$
$$q_1(45) = \max\{0.5,0.75\} = 0.75,$$
$$q_1(30) = 0.5,$$
$$q_1(0) = 0,$$
$$q_2(75) = \max\{0.7,0.72,0.496\} = 0.72,$$
$$q_2(45) = \max\{0.8q_1(30),0.96q_1(15)\} = \max\{0.4,0\} = 0.4,$$
$$q_2(15) = 0,$$
$$q_3(105) = \max\{0.648,0.396,0\} = 0.648,$$

即 $x_1 = 1$,$x_2 = 2$,$x_3 = 2$,代价 100,可靠性达到 0.648.

例 8 预测某产品后4个月的需求量分别为 220,240,200,190 个单位.4 个月后生产不变趋向稳定.相邻两个月生产能力的改变,要为之付出的代价为改变量的平方的两倍,超出需求的部分也要处理,每单位处理费用耗费 12 个单位.当月的生产能力为 200 单位.问应如何安排前 4 个月的生产.

生产能力的改变要付出代价,生产过剩了也要付处理费用,所以不是需求多少就生产多少.

令 $C_k(r)$表示上个月生产能力为 r,今后 k 个月最佳生产安排的总费用,

$$C_k(r) = \min_{x\geq s}\{2(x-r)^2 + 12(x-s) + C_{k-1}(x)\},$$

其中 s 是当月的需求量,$k = 4,3,2,1$,x 为当月的产量.假定 $C_0(x) = 0$.

当 $k = 1$ 时,$C_1(r) = \min\limits_{x_4\geq 190}\{2(x_4-r)^2 + 12(x_4-190)\}$,

其中 x_4 为第 4 个月产量.令

$$y = 2(x-r)^2 + 12(x-190),$$
$$y' = 4(x-r) + 12 = 4(x-r+3).$$

当 $r\geq 193$ 时,$x_4\geq 190$,这时

$$y = 2(3)^2 + 12(r-193) = 12r - 2298,$$

当 $r < 193$ 时,$y = 2(190-r)^2$,所以

$$C_1(r)=\begin{cases}2(190-r)^2, & r<193,\\ 12r-2298, & r\geqslant 193,\end{cases}$$
$$x_4=\max\{190, r-3\}.$$

当 $k=2$ 时,

$$C_2(r)=\min_{x_3\geqslant 200}\{2(x_3-r)^2+12(x_3-200)+C_1(x_3)\},$$

其中 x_3 为第3个月的产量.

由于 $x_3\geqslant 200$,所以 $C_1(x_3)=12x_3-2298$,

$$C_2(r)=\min_{x_3\geqslant 200}\{2(x_3-r)^2+12(x_3-200)+12x_3-2298\}.$$

令
$$y=2(x-r)^2+12(x-200)+12x-2298,$$
$$y'=4(x-r)+24=4(x-r+6),$$

所以
$$C_2(r)=\begin{cases}6(4r-795), & r\geqslant 206,\\ 2(200-r)^2+102, & r<206,\end{cases}$$
$$x_3=\max\{200, r-6\}.$$

当 $k=3$ 时,

$$C_3(r)=\min_{x_2\geqslant 240}\{2(x_2-r)^2+12(x_2-240)+C_2(x_2)\}$$
$$=\min_{x_2\geqslant 240}\{2(x_2-r)^2+12(x_2-240)+6(4x_2-795)\},$$

其中 x_2 为第2个月的产量. 类似办法可得

$$C_3(r)=\begin{cases}36r-7812, & r\geqslant 249,\\ 2(240-r)^2+990, & r<249,\end{cases}$$
$$x_2=\max\{240, r-9\}.$$

当 $k=4$ 时,

$$C_4(200)=\min_{x_1\geqslant 220}\{2(x_1-200)^2+2(x_1-220)+C_3(x_1)\}$$
$$=\begin{cases}\min\limits_{x_1\geqslant 220}\{2(x_1-200)^2+12(x_1-220)+36x_1-7812\}, & x_1\geqslant 249,\\ \min\limits_{x_1\geqslant 220}\{2(x_1-200)^2+12(x_1-220)+2(240-x_1)^2+990\}, & 220\leqslant x_1<249.\end{cases}$$

由于 $x_1\geqslant 220$ 且 $220<249$,所以 $C_3(x_1)$有两个方案可供选择. 只能选择其中一个. 可通过验证证明应选

$$C_3(x_1)=2(240-x_1)^2+990,$$
$$y=2(x_1-200)^2+12(x_1-220)+2(240-x_1)^2+990,$$
$$y'_{x_1\geqslant 220}\geqslant 0,\quad y|_{x_1=220}=2590,$$

所以
$$C_4(220)=2590.$$

即最佳的生产安排为

第一个月 $x_1=220$, 第二个月 $x_2=240$,

第三个月 $x_3=234$, 第四个月 $x_4=231$.

5　搜索技术

5.1　概　　述

前面介绍了若干算法，主要是针对离散对象的问题．然而许多很重要的实际问题，缺少有效的方法，最后只好诉之搜索，也就是穷举．穷举也是一种办法．在这一章里将介绍搜索技术，目的在于尽可能缩小搜索空间、提高效率．

例如一个4×4的棋盘，有4个棋子布在棋盘的格子里，要求每行每列都有一个且仅有一个棋子，而且两两不在一对角线上．

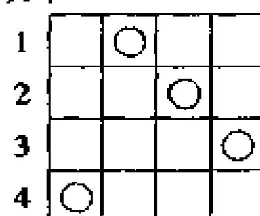


图5-1

每行每列有一个且仅有一个棋子的布局对应于1,2,3,4的某一排列．如图5-1，第1行的棋子在第2格，第2行的棋子在第3格，第3行的棋子在第4格，第4行的棋子在第1格，故对应一排列

2 3 4 1.

当然这个布局不符合要求，因3个棋子在一对角线上．

容易想到将所有的排列都罗列出来，逐个地检查，排除不符合要求的，最后便得到问题的解答．这就是穷举法，也叫做强制搜索法．

实际上大可减少搜索的对象．如图5-2所示，

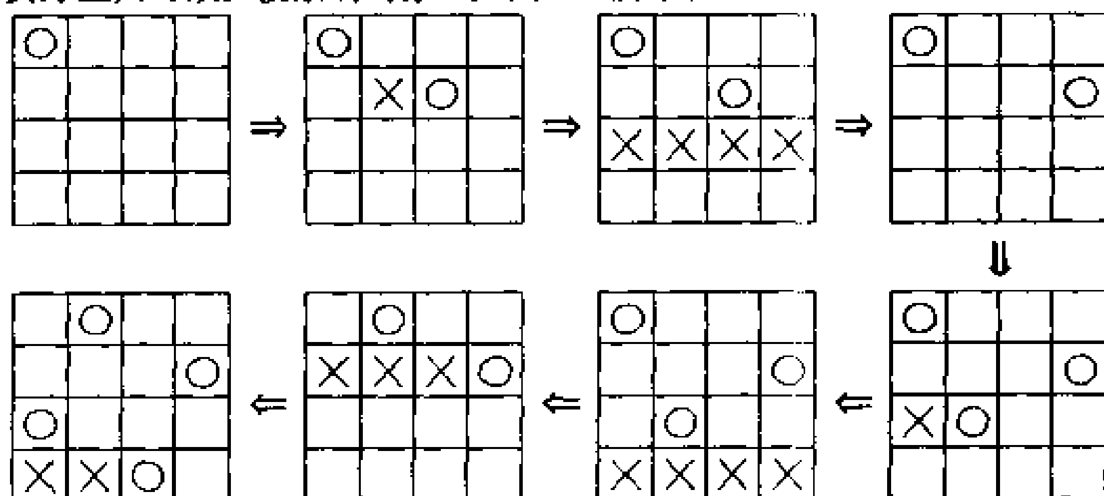


图5-2

图(a)表明第1行棋子布在第1格子；图(b)说明第2行的第1、2格不合格(x是不符合要求的表示)，故第2行布在第3格；图(c)说明第3行所有的格子都不符合要求，应退回到第2行；图(d)表示第2行棋子改布在第4格；图(e)说明在第2行棋子改布到第4格后，在第3行棋子只能布在第2格；图(f)表明在图(e)的状态第4行找不到布子的格子，退回到图(e)；图(g)表明第3行找不到可以布子的格子，应

退回第2行;图(h)表明第2行已无路可走,退回第1行,改布棋子到第2格,结果得一合符要求的布局:2 4 1 3.

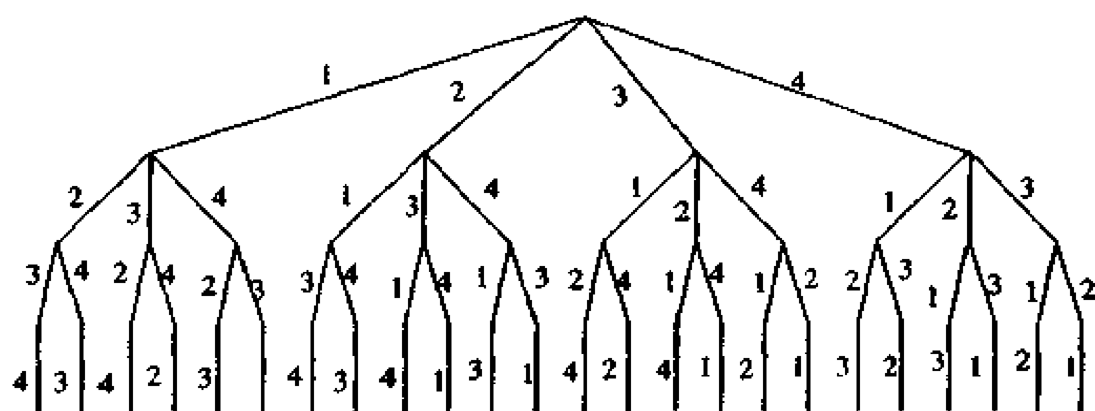


图 5-3

这例子说明,一种策略,只要前面存在前进的道路,便一往无前;否则退一步,继续寻找道路.原来的搜索空间是一棵高度为4的树(图5-3).强行搜索实际为对24个叶子的遍历.前面介绍的搜索要领可以概括为“向前走,碰壁回头”.也可以形象地用图5-4表示“回头”时剪去以下的树枝和叶子,虚线表示“碰壁回头”的返回路线.

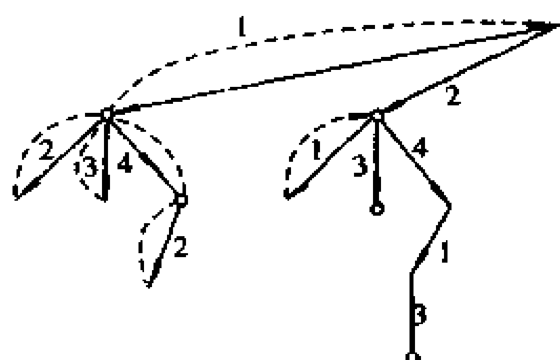


图 5-4

5.2 无向图的DFS算法

设 $G=(V,E)$ 是连通的简单无向图,要求从图的一顶点出发访问各顶点及所有的边.DFS算法如下:

步1 $T\leftarrow\varnothing, B\leftarrow\varnothing, i\leftarrow 1, \forall v\in V$ 作
$[F(v)\leftarrow 0, N(v)\leftarrow 0]$.

步2 任选一顶点 $r\in V$,作
$N(v)\leftarrow i, v\leftarrow r$.

步3 若和 r 关联的所有边均已给通过的标志则转步6,否则转步4.

步4 任选一条未标志的边 (v,w),给 (v,w) 以标志.

步5 若 $N(w)=0$,则作
$[i\leftarrow i+1, N(w)\leftarrow i, T\leftarrow T\cup\{(v,w)\}, F(w)\leftarrow v, v\leftarrow w$,转步3],
否则作[$B\leftarrow B\cup\{(v,w)\}$,转步3].

步6 若 $F(v)\neq 0$,则作
[$v\leftarrow F(v)$,转步3],
否则输出 T、B,结束.

举一例子便可对算法有较直观的理解．对图 5-5(a)从 A 点出发应用 DFS 算法，过程见图 5-5(b)．

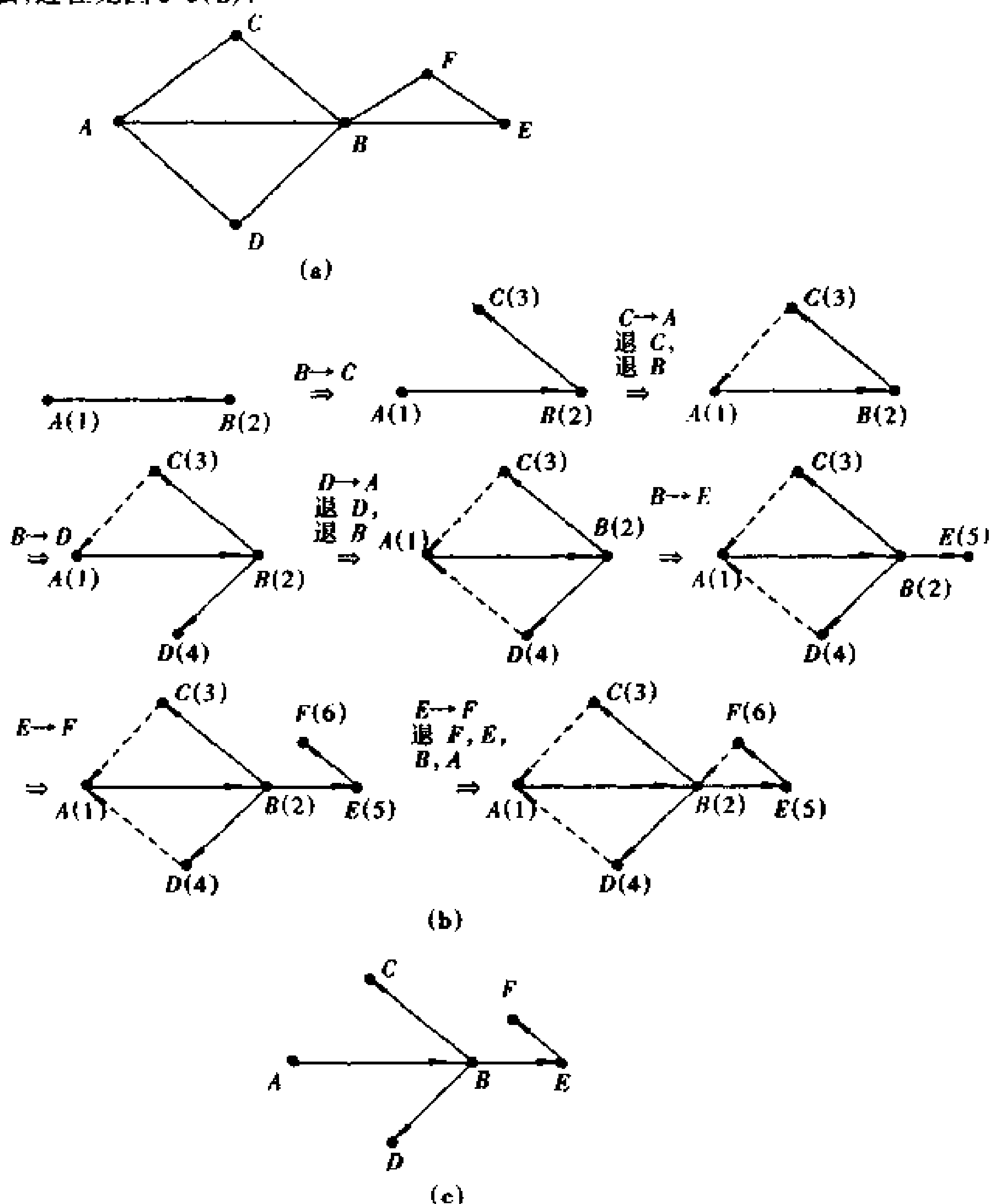


图 5-5

图 5-5(c)是 DFS 树．

图中各顶点括弧里的数 $N(\ \ )$为 DFS 访问各顶点的序数，也就叫做 DFS 数．带箭头的边构成的树称为 DFS 树，即算法中的 T．虚线为返回边即 B．算法中 $F(v)$记录 v 的"父亲"顶点，以备后退用．

5.3　有向图的 DFS 算法

设 $G=(V,E)$是有向图，有向图的 DFS 算法和无向图大致相同，只是有向图的情况复杂一些．无向图的边通过 DFS 分成 DFS 树 T 和返回边 B．有向图则除了树 T 外，分成向前边 F、后退边 B 及横跨边 C．

向前边(u,v)，即有向边(u,v)不属于 T，但 v 点是 u 点的后裔顶点，故 $N(u) < N(v)$．

后退边(u,v)，u 是 v 的后裔顶点，$N(u) > N(v)$．

横跨边(u,v)满足 $N(u) > N(v)$，但 u 不是 v 的后裔顶点，即从 v 沿 DFS 不能到达 u．

特别应指出的是，连通图的 DFS 树并不一定是一棵树，可能是"森林"．

DFS 算法：

步 1　$T \leftarrow \varnothing, F \leftarrow \varnothing, B \leftarrow \varnothing, C \leftarrow \varnothing, i \leftarrow 1$,
　　$\forall v \in V$　作$[M(v) \leftarrow 0, F(v) \leftarrow 0]$.

步 2　任选一点 r，满足条件 $M(r)=0$,
　　$N(r) \leftarrow i, M(r) \leftarrow 1, v \leftarrow r$.

步 3　若通过 v 发出的所有边均已通过检查则作
　　$[M(v) \leftarrow 2$，转步 5$]$.

步 4　任选一条未通过的边(v,w)，给(v,w)以通过标志．

(4-1) 若 $M(w)=0$　则作
　　$\{i \leftarrow i+1, N(w) \leftarrow i, T \leftarrow T \cup \{(v,w)\}$,
　　$M(w) \leftarrow 1, F(w) \leftarrow v, v \leftarrow w$，转步 3$]$.

(4-2) 若 $M(v)=2$，但 $N(w) > N(v)$，则作
　　$[F \leftarrow F \cup \{(v,w)\}$，转步 3$]$.

(4-3) 若 $M(w)=1$　但 $N(w) < N(v)$，则作
　　$[B \leftarrow B \cup \{(v,w)\}$，转步 3$]$.

(4-4) 若 $M(v)=2$，但 $N(w) < N(v)$，则作
　　$[C \leftarrow C \cup \{(v,w)\}$，转步 3$]$.

步 5　若 $F(v) \neq 0$，则作
　　$[v \leftarrow F(v)$，转步 3$]$，

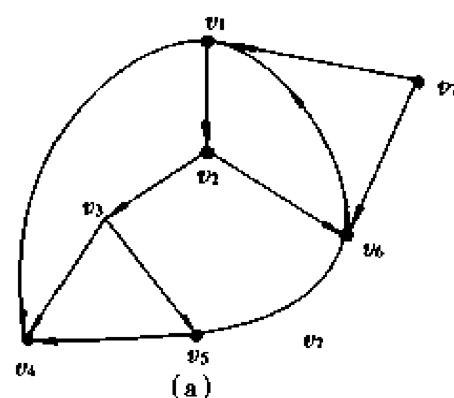
否则转步 6.

步 6　若所有的 $v \in V, M(v) \neq 0$，则结束，否则作
　　$[i \leftarrow i+1$，转步 2$]$，

其中

$$M(v)=\begin{cases}0, & v\text{ 未访问},\\1, & \text{已访问未退出},\\2, & \text{已退出}.\end{cases}$$

例1



(a)

1→2 ⇒

2→3 ⇒

3→3 ⇒

退 3
3→5 ⇒

5→4
退 5
退 3,退 2 ⇒

2→6 ⇒

6→1
退 6 ⇒

6→5
退 6,2,1 ⇒

1→4
退 1 ⇒

7→1
退 1 ⇒

7→6
退 7 ⇒

(b)

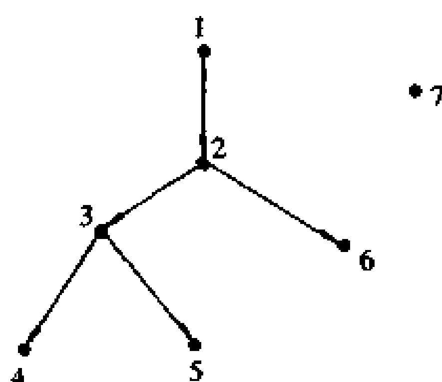


图 5-6

5.4　BFS 算法

BFS 是 Breadth First Search 的缩写，意即广度优先搜索算法，和 DFS 相对应，DFS 是“向前走，碰壁回头”，只要前面还走得通，继续向前，BFS 则不然．典型的一个例子如图 5-7 所示．

B	*H*	*C*
A		*D*
G	*F*	*E*

图 5-7

A	*B*	*C*
H		*D*
G	*F*	*E*

图 5-8

3×3 棋盘布了各种棋子，空格可以和任一邻近的棋子换位，所以可以自由游动．要求通过游动，使棋盘改变布局为图 5-8.

BFS 搜索算法即假定空格移动次序是向上、下、左、右换位，如图 5-9.

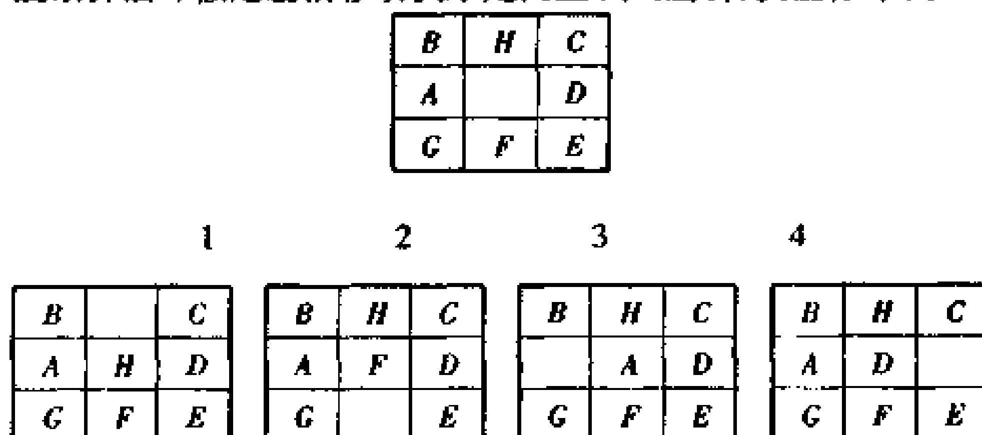


图 5-9

进而从左向右继续以上的步骤，直到出现所要求的布局为止．如何利用 BFS 寻求解答留给读者自己来完成．

5.5 α-β剪枝术

举例说明如下．现有 n 根火柴，A和B两人先后从中取1或2根，但不能不取，也不能取多于2根．最后将所有火柴取走者为胜利者．例如 $n=7$，用⑦(方框)表示轮到A取时的状态，⑥表示轮到B时的状态．故有图5-10.

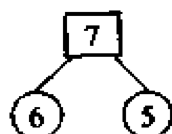


图5-10

即A取走一根，B便面对⑥状态，A取两根，B便面临⑤状态．

图5-11中□和○外面的±1，是自下而上回溯的．□是A作决策，志在于胜，○为B作决策，目的在于要A输．用1表示A胜，-1表示A败．故自下而上回溯时，状态□的值取两个儿子节点值的最大者．○状态则取其两个儿子节点值的最小者．

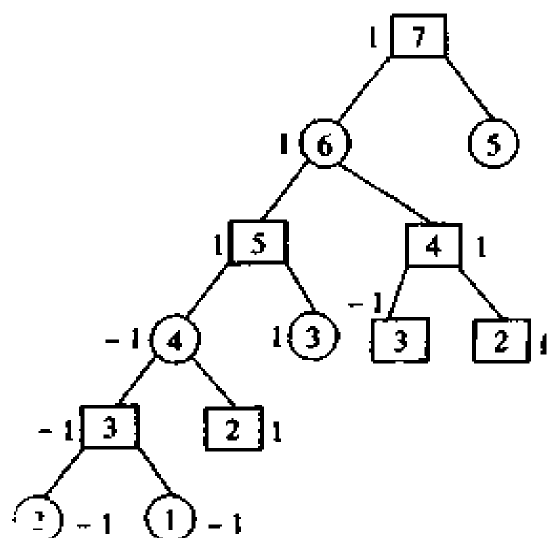


图5-11

例如④的两个儿子一取1，另一为-1．故B作决策时应让A进入-1状态．又如⑤(方框)状态的两儿子节点，一取1，另一取-1．A的决策应使B进入1的状态，即进入③状态．

图中⑦(方框)的左儿子为1，故右儿子不必搜索了．其实若进入⑤状态，A必输．

5.6 流动推销员问题的分支定界法

分支定界法是十分有效的一种搜索技术，用途很广，举例介绍方法如下：

例 2　对称型流动推销员问题.已知距离矩阵是对称的,

$$D = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} \begin{bmatrix} \infty & 12 & 1 & 16 & 2 \\ 12 & \infty & 22 & 1 & 3 \\ 1 & 22 & \infty & 9 & 7 \\ 16 & 1 & 9 & \infty & 6 \\ 2 & 3 & 7 & 6 & \infty \end{bmatrix} = [d_{ij}]_{5\times 5}$$
$$\quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$$

取距离最短的 5 条边

$$d_{13} + d_{24} + d_{15} + d_{25} + d_{45} = 13,$$

这 5 条边由于下标 5 出现了 3 次,故不是一回路,用

$$\begin{pmatrix} 13 & 24 & 15 & 25 & 45 \\ & & 13 & & \end{pmatrix}$$

表示这状态．由于下标 5 过多,考虑排除其中 d_{15},代以最短的边 35,结果得状态

$$\begin{pmatrix} 13 & 24 & 25 & 45 & 35 \\ & & 18 & & \end{pmatrix}.$$

若考虑保留 d_{15},排除 d_{25}结果得状态

$$\begin{pmatrix} 13 & 24 & 15 & 45 & 35 \\ & & 17 & & \end{pmatrix}.$$

若考虑保留 d_{15}、d_{25},排除 d_{45}得状态

$$\begin{pmatrix} 13 & 24 & 15 & 25 & 35 \\ & & 14 & & \end{pmatrix}.$$

若考虑保留 d_{15}、d_{25},排除 d_{45}、d_{35},结果得状态

$$\begin{pmatrix} 13 & 24 & 15 & 25 & 34 \\ & & 16 & & \end{pmatrix}.$$

$$v_1 \to v_3 \to v_4 \to v_2 \to v_5 \to v_1$$

是一条最短回路,总长为 16.

搜索过程用图 5-12 表示,其中$\overline{15}$表示排除 d_{15},它的另一面为保留 d_{15}.余此类推．

$\begin{pmatrix} 13 & 24 & 15 & 25 & 34 \\ & & 16 & & \end{pmatrix}$状态是否最优?这棵搜索树的树叶 $\begin{pmatrix} 13 & 24 & 25 & 45 & 35 \\ & & 17 & & \end{pmatrix}$和$\begin{pmatrix} 13 & 24 & 15 & 45 & 35 \\ & & 17 & & \end{pmatrix}$为两种状态,虽然不是回路,但它的界已超过 16,故没有搜索的价值．所以

$$\begin{pmatrix} 13 & 24 & 15 & 25 & 34 \\ & & 16 & & \end{pmatrix}$$

是最优解．

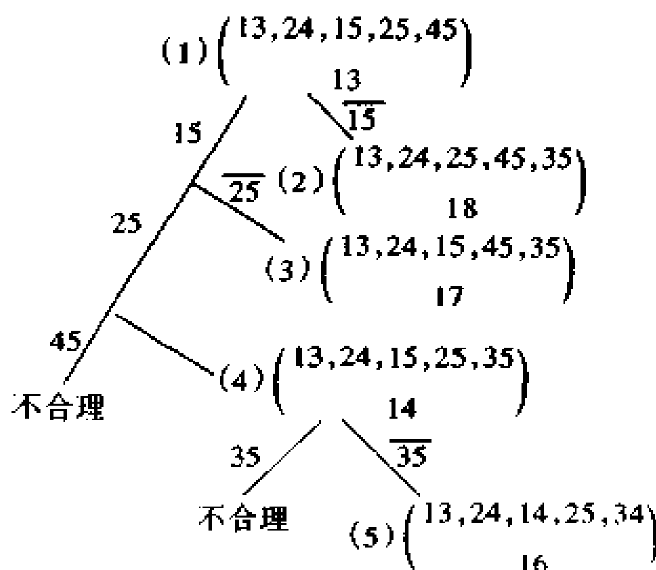


图 5-12

例 3 非对称型流动推销员问题.

所谓非对称型指的是距离矩阵不是对称矩阵,即 d_{ij} 不一定等于 d_{ji}.

已知

$$\boldsymbol{D}=\begin{array}{c} v_1\\ v_2\\ v_3\\ v_4\\ v_5 \end{array}\begin{bmatrix} \infty & 24 & 34 & 14 & 15\\ 19 & \infty & 20 & 9 & 6\\ 7 & 9 & \infty & 6 & 8\\ 23 & 10 & 22 & \infty & 7\\ 20 & 8 & 11 & 20 & \infty \end{bmatrix}.$$
$$\quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$$

为了便于理解,假定 $\boldsymbol{D}$ 是旅费矩阵,求旅费最少的回路.

对 $\boldsymbol{D}$ 的每行用它的最小元素来减,各列也一样. 比如 $\boldsymbol{D}$ 有

$$\begin{bmatrix} \infty & 24 & 34 & 14 & 15\\ 19 & \infty & 20 & 9 & 6\\ 7 & 9 & \infty & 6 & 8\\ 23 & 10 & 22 & \infty & 7\\ 20 & 8 & 11 & 20 & \infty \end{bmatrix}\begin{array}{l} 14\\ 6\\ 6\\ 7\\ 8 \end{array} \Rightarrow \begin{bmatrix} \infty & 10 & 20 & 0 & 1\\ 13 & \infty & 14 & 3 & 0\\ 1 & 3 & \infty & 0 & 2\\ 16 & 3 & 15 & \infty & 0\\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}_{41}$$
$$\begin{array}{ccccc} 1 & & 3 & & \end{array}$$

$$\Rightarrow \begin{bmatrix} \infty & 10 & 17 & 0 & 1\\ 12 & \infty & 11 & 3 & 0\\ 0 & 3 & \infty & 0 & 2\\ 15 & 3 & 12 & \infty & 0\\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}_{45} = \boldsymbol{D}_0$$

矩阵右下角的 41 和 45 分别是

$$14+6+6+7+8=41,$$
$$41+1+3=45,$$

由于回路对各顶点进出都是一次．各行用它的最小元素来减,比如第1行减以14,表示从 v_1 发出的所有旅费一律降价13单位．又如第1列减以1,表示进入 v_1 的旅费一律降价1单位．所以原来问题的最佳回路也是后面矩阵 D_0 的最佳回路．矩阵 D_0 的特点是每行都有0元素,每列也都有0元素．如若能找到5个0元素正好是在不同行不同列,这5个0元素给出了问题的解．

$$
\boldsymbol{D}_0 = \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ \ \end{array}
\begin{array}{c}
\left[\begin{array}{ccccc}
\infty & 10 & 17 & 0^* & 1 \\
12 & \infty & 11 & 3 & 0 \\
0 & 3 & \infty & 0 & 2 \\
15 & 3 & 12 & \infty & 0 \\
11 & 0 & 0 & 12 & \infty
\end{array}\right]_{45} \\
\begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array}
\end{array} .
$$

从矩阵 $D_0 v_1$ 下一站自然选择 v_4, $d_{14}^* = 0$. 由于从 v_1 出发仅一次, v_4 仅进入一次．将第1行第4列划去,并将 d_{41} 改为 ∞,得 D_1,

$$
\boldsymbol{D}_1 = \begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \\ \ \end{array}
\begin{array}{c}
\left[\begin{array}{cccc}
12 & \infty & 11 & 0^* \\
0 & 3 & \infty & 2 \\
\infty & 3 & 12 & 0 \\
11 & 0 & 0 & \infty
\end{array}\right]_{45} \\
\begin{array}{cccc} v_1 & v_2 & v_3 & v_5 \end{array}
\end{array} ,
$$

即排除出现 $v_1 \to v_4 \to v_1$ 的可能．v_2 出发走向 v_5,同样办法处理有

$$
\begin{array}{c} v_3 \\ v_4 \\ v_5 \\ \ \end{array}
\begin{array}{c}
\left[\begin{array}{ccc}
0 & 3 & \infty \\
\infty & 3 & 12 \\
11 & \infty & 0
\end{array}\right]_{45} \\
\begin{array}{ccc} v_1 & v_2 & v_3 \end{array}
\end{array} ,
$$

v_4 行减去3得

$$
\begin{array}{c} v_3 \\ v_4 \\ v_5 \\ \ \end{array}
\begin{array}{c}
\left[\begin{array}{ccc}
0 & 3 & \infty \\
\infty & 0 & 9 \\
11 & \infty & 0
\end{array}\right]_{48} \\
\begin{array}{ccc} v_1 & v_2 & v_3 \end{array}
\end{array} ,
$$

从而可得 $v_1 \to v_4 \to v_2 \to v_5 \to v_3 \to v_1$ 是旅费最少回路,旅费48单位．

与 $v_1 \to v_4$ 相反,排除 $v_1 \to v_4$ 的可能即在 D_0 矩阵中让 $d_{14} = \infty$,并将第1行元素减去最小元素1,得矩阵

$$
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ \ \end{array}
\begin{array}{c}
\left[\begin{array}{ccccc}
\infty & 9 & 16 & \infty & 0 \\
12 & \infty & 11 & 3 & 0 \\
0 & 3 & \infty & 0 & 2 \\
15 & 3 & 12 & \infty & 0 \\
11 & 0 & 0 & 12 & \infty
\end{array}\right]_{46} \\
\begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array}
\end{array} .
$$

类似有 $v_2 \to v_5$ 的反面, $v_2 \to v_5$ 被排除,即令 $d_{25} = \infty$. 搜索全过程如下：

$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} \infty & 10 & 17 & 0^* & 1 \\ 12 & \infty & 11 & 3 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}_{45} \quad (\text{列: } v_1\ v_2\ v_3\ v_4\ v_5)$$

$v_1 \to v_4$:

$$\begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} 12 & \infty & 11 & 0^* \\ 0 & 3 & \infty & 2 \\ \infty & 3 & 12 & 0 \\ 11 & 0 & 0 & \infty \end{bmatrix}_{45} \quad (\text{列: } v_1\ v_2\ v_3\ v_5)$$

$v_1 \not\to v_4$:

$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} \infty & 9 & 16 & \infty & 0^* \\ 12 & \infty & 11 & 3 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}_{46} \quad (\text{列: } v_1\ v_2\ v_3\ v_4\ v_5)$$

$v_2 \to v_5$:

$$\begin{array}{c} v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} 0 & 3 & \infty \\ \infty & 0 & 9 \\ 11 & \infty & 0 \end{bmatrix}_{48} \quad (\text{列: } v_1\ v_2\ v_3)$$

$v_2 \not\to v_5$:

$$\begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} 1 & \infty & 0 & \infty \\ 0 & 3 & \infty & 2 \\ \infty & 3 & 12 & 0 \\ 11 & 0 & 0 & \infty \end{bmatrix}_{56} \quad (\text{列: } v_1\ v_2\ v_3\ v_5)$$

$v_4 \to v_2$:

$$\begin{array}{c} v_3 \\ v_5 \end{array}\begin{bmatrix} 0 & \infty \\ 11 & 0 \end{bmatrix}^{*}_{48} \quad (\text{列: } v_1\ v_3)$$

$v_4 \not\to v_2$:

$$\begin{array}{c} v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} 0 & 0 & \infty \\ \infty & \infty & 0 \\ 11 & \infty & 0 \end{bmatrix}_{60} \quad (\text{列: } v_1\ v_2\ v_3)$$

由 $v_1 \to v_4$ 左分支可见最优的回路为

$$v_1 \to v_4 \to v_2 \to v_5 \to v_3 \to v_1$$

旅费为 48. 但排除 $v_1 \to v_4$ 的右分支,估界 $46 < 48$,所以还须继续搜索.

$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{bmatrix} \infty & 9 & 16 & \infty & 0^* \\ 12 & \infty & 11 & 3 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}_{46} \quad (\text{列: } v_1\ v_2\ v_3\ v_4\ v_5)$$

$v_1 \to v_5$ / \ $v_1 \not\to v_5$

$$
\begin{array}{c}
\begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{bmatrix} 9 & \infty & 8 & 0 \\ 0 & 3 & \infty & 0 \\ 15 & 3 & 12 & \infty \\ \infty & 0 & 0 & 12 \end{bmatrix}_{49} \\
\quad v_1 \quad v_2 \quad v_3 \quad v_4
\end{array}
\qquad
\begin{array}{c}
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{bmatrix} \infty & 0 & 7 & \infty & \infty \\ 12 & \infty & 11 & 3 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}_{55} \\
\quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5
\end{array}
$$

由于 49 和 55 均超过 48,即估界至少都在 48 之上,故无搜索的价值．搜索结束．

5.7　同顺序加工任务安排

假定有 J_1、J_2、J_3、J_4 四个项目需要加工,加工的顺序相同:$m_1 \to m_2 \to m_3$,加工的时间矩阵

$$
T = \begin{array}{c} J_1 \\ J_2 \\ J_3 \\ J_4 \end{array}
\begin{bmatrix} 5 & 7 & 9 \\ 10 & 5 & 4 \\ 9 & 7 & 5 \\ 5 & 8 & 10 \end{bmatrix}.
\quad (\text{列: } m_1 \ m_2 \ m_3)
$$

t_{ij}为 J_i 任务通过 m_j 工序所需的时间．

假如加工顺序为 $J_2 \to J_3 \to J_1 \to J_4$,则加工过程的时间进程如图 5-13 所示．

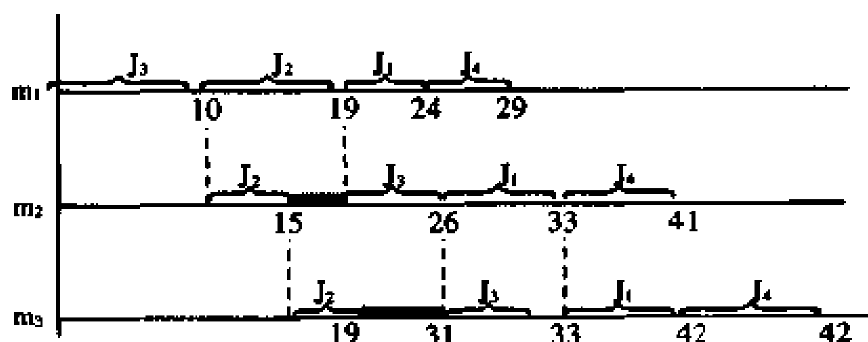


图 5-13

图中 ////////// 表示机器空闲等任务．从开始到结束共 52 单位时间．任务安排的方案不同结果可能差别很大．

下面介绍用分支定界法找最佳方案的方法．关键是对时间的估界．

(1) 从 J_i 最先开始的时间估界:

$$
t_{i1} + \sum_{j=1}^{4} t_{j2} + \min_{k \neq i} \{ t_{k3} \}.
$$

(2) 从 J_i 开始继以 J_j 的时间估界

$$t_{i1} + t_{j1} + \sum_{k \neq i} t_{k2} + \min_{k \neq i,j} | t_{k3} | .$$

(3) $J_i \to J_j \to J_k$ 的时间估界

$$t_{i1} + t_{j1} + t_{k1} + \sum_{h \neq i,j} t_{h2} + t_{l3}, \quad l \neq i,j,k.$$

例如 J_1、J_2、J_3、J_4 开始估的界如图 5-14.

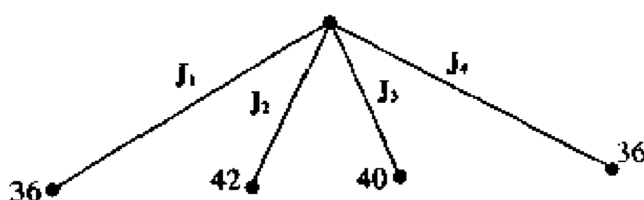


图 5-14

可见以 J_1、J_4 开始界最低,先从 J_1、J_4 开始搜索．搜索过程如图 5-15 所示:$J_1 \to J_4 \to J_3$ 估界 36 为最低．$J_1 \to J_4 \to J_3 \to J_2$ 的加工时间为 40.

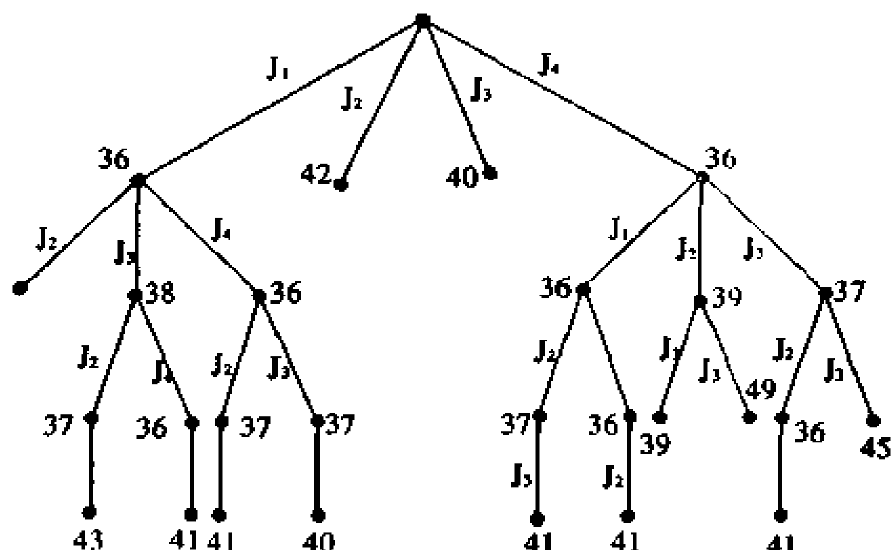


图 5-15

J_2 和 J_3 开始的界都超过或等于 40. 故没有搜索价值而被剪掉．凡是界小于 40 的都需要进一步继续搜索．

6　FFT 并行算法与脉动阵列

6.1　并行计算概念

随着计算机科学的发展,并行计算异军突起．在这以前数学计算都是串形地进行,人的大脑活动也是串形,虽然同时处理若干事务,但每个时间间隔的思维仍

然是串形的．并行计算顾名思义由若干个处理器或计算机同时工作，以提高工作效率．毫无疑问一个人做的工作如能分由几个人一齐做，时间将会缩短．特别是大型的计算问题，如天气预报、空间技术中的图像传输等，要求适时，所以要求计算时间短，否则就是算出来也将失去意义．在这里并行计算技术起到关键作用．这一章里将介绍其中比较重要的FFT算法和脉动阵列．

并行计算要克服人们串行思维的格式．以递推关系为例，

$$a_i = b_i a_{i-1} + c_i,\quad a_0 = d,\quad i = 1,2,\cdots,n$$

递推产生的序列 $a_1, a_2, \cdots, a_n$ 能否并行处理呢？由于

$$a_i = b_i a_{i-1} + c_i,\qquad a_0 = d, \tag{6-1}$$

$$a_{i-1} = b_{i-1} a_{i-2} + c_{i-1},$$

所以

$$\begin{aligned} a_i &= b_i(b_{i-1}a_{i-2} + c_{i-1}) + c_i = b_i b_{i-1} a_{i-2} + b_i c_{i-1} + c_i \\ &= b_i^{(1)} a_{i-2} + b_i^{(1)},\quad i = 2,3,\cdots,n, \end{aligned} \tag{6-2}$$

其中

$$b_i^{(1)} = b_i b_{i-1},\quad c_i^{(1)} = b_i c_{i-1} + c_i.$$

$b_i^{(1)}$ 和 $c_i^{(1)}$，$i = 2,3,\cdots,n$，也可以并行计算．

如果说从(6-1)式可以计算得 a_1，已知 a_0、a_1 则从(6-2)式可以同时计算 a_2、a_3．这里同时指的是"并行"计算得到 a_2、a_3．

继续以上的步骤可得

$$a_i = b_i^{(l)} a_{i-2^l} + c_i^{(l)},\quad i = 2^l, 2^l+1, \cdots, n, l = 1,2,\cdots,\mathrm{lb}n.$$

例如 $l = 2$，

$$a_i = b_i^{(2)} a_{i-4} + c_i^{(2)},\quad i = 4,5,\cdots,n.$$

从已知 a_0, a_1, a_2, a_3 可并行计算得 $a_4, a_5, a_6, a_7, \cdots$

6.2　FFT

FFT是Fast Fourier Transform的缩写，意即快速傅里叶变换．随着空间技术的突飞猛进地发展，从卫星上拍下的大量照片可以通过电讯号传送回地面．若考虑将照片分成 $n \times m$ 个格子，将每个格子上光的强弱数值化送回地面，但数据量之大使得这方法实际上无法实现．若将图片化作二维的傅里叶级数，由于送回的傅氏系数的高频部分的系数大量为零，所以传输容易，因此达到压缩数据的目的．地面上收到傅氏系数后，可重新还原原来的图像．为了加快运算速度，"快速傅里叶变换"应运而生．后面将看到并行计算在FFT中起到关键的作用，是并行计算成功的典型．

6.2.1　预备定理

定理1　设 r 和 m 都是正整数，则

$$\sum_{k=0}^{n-1} e^{2k\pi i(r-m)/n} = \begin{cases} n, & r \equiv m \pmod{n}, \\ 0, & r \not\equiv m \pmod{n}. \end{cases}$$

$r \equiv m \pmod{n}$，则 $e^{2\pi i(r-m)/n} = 1$．

$r \neq m\ (\mathrm{mod}\, n)$，等式左端是等比级数，公比为 $e^{2\pi i(r-m)/n}$. 通过直接计算容易证此结论．

定理 2 若数列 $x(0), x(1), \cdots, x(n-1)$ 和数列 $y(0), y(1), \cdots, y(n-1)$ 满足关系

$$y(k) = \frac{1}{n}\sum_{j=0}^{n-1} x(j) e^{-2\pi ijk/n}, \quad k = 0,1,2,\cdots,n-1, \tag{6-3}$$

则有

$$x(j) = \sum_{k=0}^{n-1} y(k) e^{2\pi ijk/n}, \quad j = 0,1,2,\cdots,n-1. \tag{6-4}$$

若将(6-3)式看作是关于 $x(0), x(1), \cdots, x(n-1)$ 的方程组，则(6-4)式给出(6-3)式的解．反之，若将(6-4)式看作是 $y(0), y(1), \cdots, y(n-1)$ 的代数方程组，则(6-3)式便是它的解．

只要将它代入即可验证．例如将(6-4)式代入(6-3)式，k 是常数，得

$$\begin{aligned}\frac{1}{n}\sum_{j=0}^{n-1} x(j) e^{-2\pi ijk/n} &= \frac{1}{n}\sum_{j=0}^{n-1}\sum_{r=0}^{n-1} y(r) e^{-2\pi ij(k-r)/n} \\ &= \frac{1}{n}\sum_{r=0}^{n-1} y(r) \sum_{j=0}^{n-1} e^{-2\pi ij(k-r)/n},\end{aligned}$$

由定理 1 及 $r \neq k$ 时 $\sum_{j=0}^{n-1} e^{2\pi ij(k-r)/n} = 0$，所以

$$\frac{1}{n}\sum_{j=0}^{n-1} x(j) e^{-2\pi ijk/n} = y(k), \quad k = 0,1,2,\cdots,n-1$$

定理 2 有着极其重要的含义，即将(6-3)式看作对序列 $x(0), x(1), \cdots, x(n-1)$ 进行傅里叶变换，得 $y(0), y(1), \cdots, y(n-1)$，则(6-4)式是傅里叶变换的逆变换．

已知 $x(0), x(1), \cdots, x(n-1)$，作离散的傅里叶变换得 $y(0), y(1), \cdots, y(n-1)$，相当于从已知函数 $x(t)$，在 $(0, l)$ 区间上作傅里叶级数展开，得傅里叶系数

$$c_k = \frac{1}{l}\int_0^l x(\tau) e^{-2k\pi i\tau/l} d\tau, \quad k = 0, \pm 1, \pm 2, \cdots,$$

使得

$$x(t) \sim \sum_{k=-\infty}^{\infty} c_k e^{2k\pi it/l}$$

读者将看到已知 $x(t)$ 在 $(0, l)$ 上的 n 个等分点 $x(0), x(1), \cdots, x(n-1)$ 计算 c_k，便导致求 $y(k), k = 0,1,2,\cdots,n-1$. 所以作离散的傅里叶变换相当用近似积分计算傅氏系数．

离散的傅里叶逆变换相当于从已知傅氏系数 $y(0), y(1), \cdots, y(n-1)$ 求 $x(0), x(1), \cdots, x(n-1)$.

6.2.2 快速算法

已知 $x(0), x(1), \cdots, x(n-1)$ 计算 $y(0), y(1), \cdots, y(n-1)$，作 $n^2 + n$ 次乘法，从 $y(0), y(1), \cdots, y(n-1)$ 求 $x(0), x(1), \cdots, x(n-1)$ 需作 n^2 次乘法．

(1) $n=2$ 时，由于 $e^{\pi i}=-1$，所以

$$x(0)=y(0)+y(1),$$
$$x(1)=y(0)-y(1),$$

或

$$\begin{bmatrix} x(0) \\ x(1) \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} y(0) \\ y(1) \end{bmatrix}.$$

实际并非作 4 次乘法，而是两次加法．如图 6-1 所示．

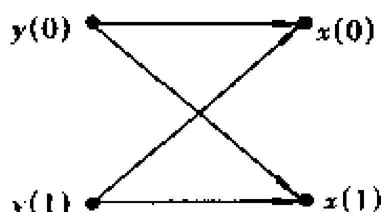


图 6-1

若作并行计算，实际上只要一步完成．

(2) $n=4$ 时，令 $w_4=e^{2\pi i/4}=e^{\pi i/2}=i$，

$$w_4^0=1,\quad w_4^1=i,\quad w_4^2=-1,\quad w_4^3=-i,$$
$$x(0)=y(0)+y(1)+y(2)+y(3),$$
$$x(2)=y(0)+y(1)w_4^2+y(2)+y(3)w_4^2,$$
$$x(1)=y(0)+y(1)w_4+y(2)w_4^2+y(3)w_4^3,$$
$$x(3)=y(0)+y(1)w_4^3+y(2)w_4^2+y(3)w_4,$$

或

$$\begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix}=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w_4^2 & 1 & w_4^2 \\ 1 & w_4 & w_4^2 & w_4^3 \\ 1 & w_4^3 & w_4^2 & w_4 \end{bmatrix}\begin{bmatrix} y(0) \\ y(1) \\ y_2(2) \\ y_2(3) \end{bmatrix}.$$

但

$$w_4^2=w_2,$$

$$\begin{bmatrix} w_4^2 & w_4^3 \\ w_4^2 & w_4 \end{bmatrix}=w_4^2\begin{bmatrix} 1 & w_4 \\ 1 & -w_4 \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ 1 & w_4^2 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & -w_4 \end{bmatrix},$$

所以

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & w_4^2 & 1 & w_4^2 \\ \hline 1 & w_4 & w_4^2 & w_4^3 \\ 1 & w_4^3 & w_4^2 & w_4 \end{array}\right]=\left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 1 & w_2 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & w_2 \end{array}\right]\left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 0 & -1 & 0 \\ 0 & w_4 & 0 & -w_4 \end{array}\right],$$

$$\begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix}=\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & w_2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & w_2 \end{bmatrix}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 2 & w_4 & 0 & -w_4 \end{bmatrix}\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix}.$$

令

$$\begin{bmatrix} z(0) \\ z(1) \\ z(2) \\ z(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -w_4 \end{bmatrix} \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix},$$

则

$$\begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & w_2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & w_2 \end{bmatrix} \begin{bmatrix} z(0) \\ z(1) \\ z(2) \\ z(3) \end{bmatrix},$$

或

$$\begin{bmatrix} x(0) \\ x(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & w_2 \end{bmatrix} \begin{bmatrix} z(0) \\ z(1) \end{bmatrix},$$

$$\begin{bmatrix} x(1) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & w_2 \end{bmatrix} \begin{bmatrix} z(2) \\ z(3) \end{bmatrix}.$$

将由 $y(0), y(1), y(2), y(3)$ 到 $x(0), x(2), x(1), x(3)$ 的过程用流程图的形式表示,则如图 6-2.

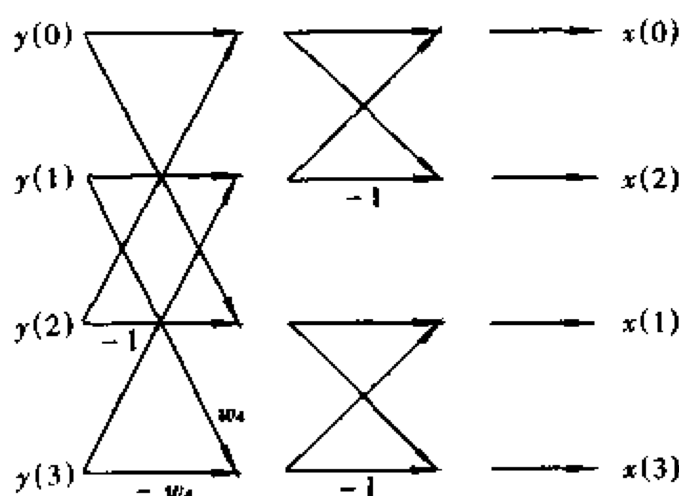


图 6-2

$n = 2^3$; $w_8 = e^{\pi i/4}$, $w_8^2 = w_4$, $w_8^4 = w_2$.

$$\begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \\ x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} = \left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_8^4 & 1 & w_8^4 & 1 & w_8^4 & 1 & w_8^4 \\ 1 & w_8^2 & w_8^4 & w_8^6 & 1 & w_8^2 & w_8^4 & w_8^6 \\ 1 & w_8^4 & w_8^4 & w_8^2 & 1 & w_8^6 & w_8^4 & w_8^2 \\ \hdashline 1 & w_8 & w_8^2 & w_8^3 & w_8^4 & w_8^5 & w_8^6 & w_8^7 \\ 1 & w_8^5 & w_8^2 & w_8^7 & w_8^4 & w_8 & w_8^6 & w_8^3 \\ 1 & w_8^3 & w_8^6 & w_8 & w_8^4 & w_8^7 & w_8^2 & w_8^5 \\ 1 & w_8^7 & w_8^6 & w_8^5 & w_8^4 & w_8^3 & w_8^2 & w_8 \end{array}\right] \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \\ y(7) \end{bmatrix}.$$

但

$$\begin{bmatrix} 1 & w_8 & w_8^2 & w_8^3 \\ 1 & w_8^5 & w_8^2 & w_8^7 \\ 1 & w_8^3 & w_8^6 & w_8 \\ 1 & w_8^7 & w_8^6 & w_8^5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w_8^4 & 1 & w_8^4 \\ 1 & w_8^2 & w_8^4 & w_8^6 \\ 1 & w_8^6 & w_8^4 & w_8^2 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & w_8 & & \\ & & w_8^2 & \\ & & & w_8^3 \end{bmatrix},$$

$$\begin{bmatrix} w_8^4 & w_8^5 & w_8^6 & w_8^7 \\ w_8^4 & w_8 & w_8^6 & w_8^3 \\ w_8^4 & w_8^7 & w_8^2 & w_8^5 \\ w_8^4 & w_8^5 & w_8^2 & w_8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w_8^4 & 1 & w_8^4 \\ 1 & w_8^2 & w_8^4 & w_8^6 \\ 1 & w_8^6 & w_8^4 & w_8^2 \end{bmatrix} \begin{bmatrix} -1 & & & \\ & -w_8 & & \\ & & -w_8^2 & \\ & & & -w_8^3 \end{bmatrix},$$

所以

$$\left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_8^4 & 1 & w_8^4 & 1 & w_8^4 & 1 & w_8^4 \\ 1 & w_8^2 & w_8^4 & w_8^6 & 1 & w_8^2 & w_8^4 & w_8^6 \\ 1 & w_8^6 & w_8^4 & w_8^2 & 1 & w_8^6 & w_8^4 & w_8^2 \\ \hdashline 1 & w_8 & w_8^2 & w_8^3 & w_8^4 & w_8^5 & w_8^6 & w_8^7 \\ 1 & w_8^5 & w_8^2 & w_8^7 & w_8^4 & w_8 & w_8^6 & w_8^3 \\ 1 & w_8^3 & w_8^6 & w_8 & w_8^4 & w_8^7 & w_8^2 & w_8^5 \\ 1 & w_8^7 & w_8^6 & w_8^5 & w_8^4 & w_8^3 & w_8^2 & w_8 \end{array}\right],$$

$$\left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & & & & \\ 1 & w_8^4 & 1 & w_8^4 & & \mathbf{0} & & \\ 1 & w_8^2 & w_8^4 & w_8^6 & & & & \\ 1 & w_8^6 & w_8^4 & w_8^2 & & & & \\ \hdashline & & & & 1 & 1 & 1 & 1 \\ & \mathbf{0} & & & 1 & w_8^4 & 1 & w_8^4 \\ & & & & 1 & w_8^2 & w_8^4 & w_8^6 \\ & & & & 1 & w_8^6 & w_8^4 & w_8^2 \end{array}\right],$$

$$\left[\begin{array}{cccc:cccc} 1 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & \mathbf{0} & & 1 \\ \hdashline 1 & & & & -1 & & & \\ & w_8 & & & & -w_8 & & \\ & & w_8^2 & & & & -w_8^2 & \\ & & & w_8^3 & & \mathbf{0} & & -w_8^3 \end{array}\right].$$

注意

$$\begin{bmatrix}1&1&1&1\\1&w_8^4&1&w_8^4\\1&w_8^2&w_8^4&w_8^6\\1&w_8^6&w_8^4&w_8^2\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&w_4^2&1&w_4^2\\1&w_4&w_4^2&w_4^3\\1&w_4^3&w_4^2&w_4\end{bmatrix},$$

所以从 $y(0),y(1),\cdots,y(7)$到 $x(0),x(1),\cdots,x(7)$的变换过程的流程图如图 6-3.

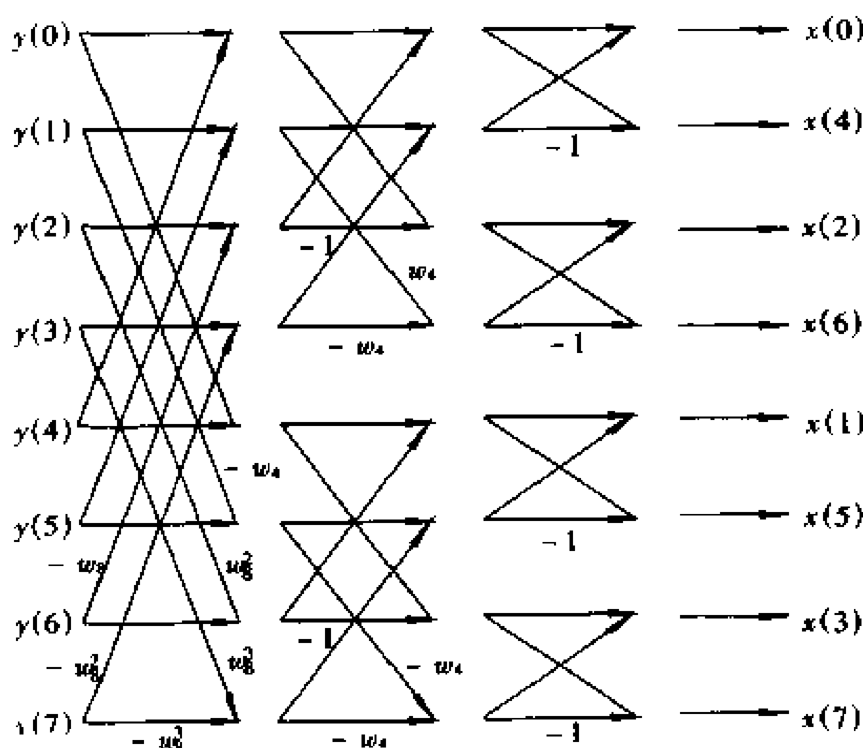


图 6-3

$x(0),x(4),x(2),x(6),x(1),x(5),x(3),x(7)$的顺序有规律可循,从 $n=1$ 时的 $x(0),x(1)$和 $n=2$ 时的 $x(0),x(2),x(1),x(3)$以及 $n=3$ 时的 $x(0),x(4),x(2),x(6),x(1),x(5),x(3),x(7)$,不难找到它的规律性.

$n=8$ 的 FFT 共用 5 次乘法,24 加法. 若用并行算法则 5 步完成,其中两次乘法,3 次加法.

从 $n=2$ 到 $n=4$,从 $n=4$ 到 $n=8$,其规律带有普遍性,也就是说可以推到 $n=2^4,2^5,\cdots$.

前面已强调说过 FFT 是并行算法的典型例子,对于 $n=2^k$,执行的步数大致为 $k=\text{lb}n$. 若考虑用一硬件设备来完成 FFT,每一操作用一简单元件来完成,基本上流程图中的点数也就是元件数。设 $n=2^k$ 时 FFT 计算装置的元件数目为 A_k,则

$$A_k=2A_{k-1}+2^k,$$

$$A_1=2.$$

于是 $A_2=2^2+2^2=2\cdot2^2, A_3=2\cdot2^3+2^3=3\cdot3^3$. 设 $A_{k-1}=(k-1)2^{k-1}$,则

$$A_k=(k-1)2^k+2^k=k2^k=n\,\text{lb}\,n,$$

即 n 个元素的 FFT 计算装置所需要的元件数目为 $n\mathrm{lb}n$. 说明 FFT 所作的加法和乘法数,即串行计算的时间复杂性为 $O(n\mathrm{lb}n)$.

具体地说设 $n=2^k$ 的乘法数,为 M_k,则有

$$M_k=2M_{k-1}+2^{k-1}-1,\quad M_1=0,$$

$$M_n=\frac{1}{2}n\mathrm{lb}n-n+1.$$

类似可得加法数为 $n\mathrm{lb}n$.

6.3　脉动阵列的并行计算装置

由 n 个处理器组成并行机,麻烦在于处理器间的连接,若导线多而且长,实际上将难于实现. 因此自然地希望处理器排成队列形式,每个处理器只和邻近的处理器相连. 每走一步是各处理器从相邻的处理器读进一个数,然后进行一个运算,再将结果写入下一个邻近的处理器,这就是"脉动阵列". 最典型的例子是矩阵乘法. 下面是一向量与矩阵相乘的例子. 例如

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix}=\begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{6-5}$$

可通过一系列的如图 6-4 所示的处理器来完成. 每个处理器有上、左和右三条输入线,以及左、右两条输出线,即将左边输入和上面输入作乘的运算,再和右边输入作加法运算,结果从左方输出线输出.

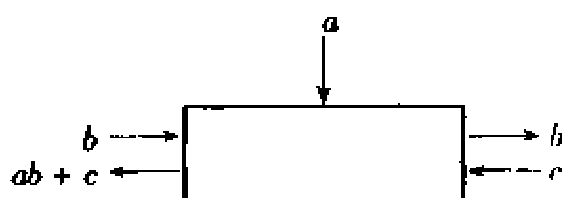
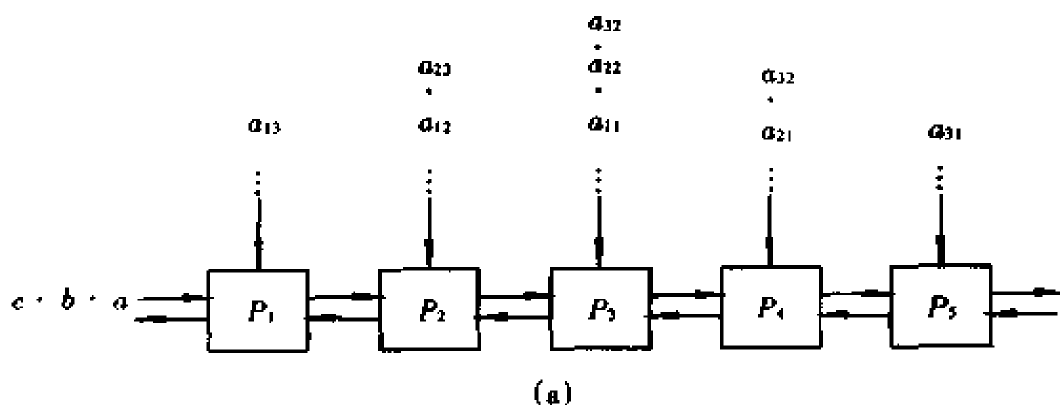


图 6-4

图 6-5 给出了计算(6-4)式的动态过程.



(a)

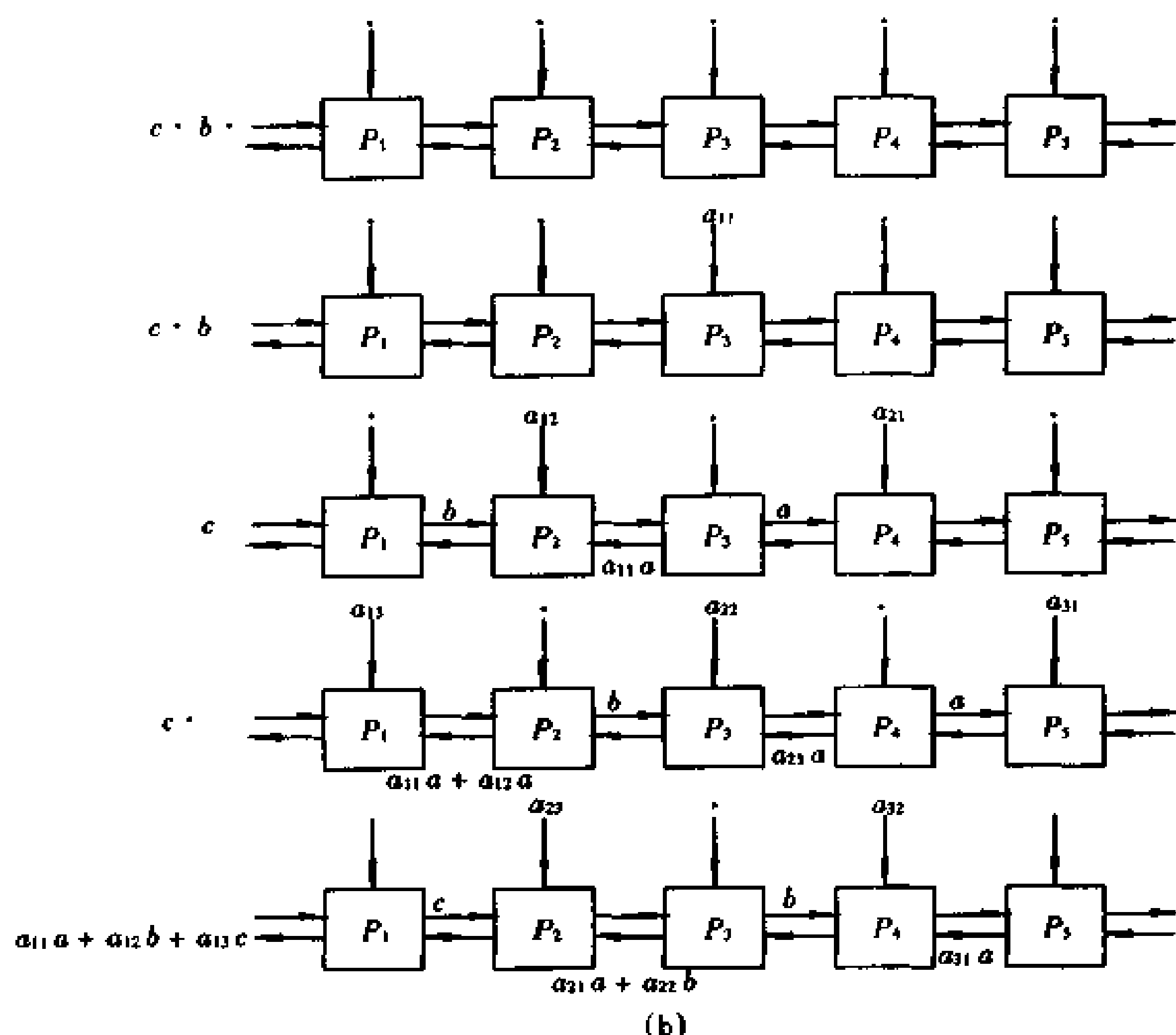


图 6-5

还可以构造用于作矩阵乘法的脉动阵列，这里从略.

7 排序与查找

7.1 排序的下界估计

本篇前面六章，分别讨论各自的算法.但这一章“排序与查找”，研究的是另一类问题.文件储存在计算机系统里，是按照它们的关键字依某种顺序排列的.“排序与查找”成为计算机科学的重要课题，还由于计算机在运行中相当多的时间是在处理这类问题，而且环境各异，要求也不尽相同，所以方法不一而足.系统地研究“排序与查找”不是这一章的目的，只拟用前面的算法分类来分析典型的排序及查找的算法.

在这一节里将先讨论排序算法的下界的估计.

所谓排序,是对 n 个元素的集合,要求将它按大小顺序排列:

$$a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n.$$

所谓排序的下界,指在最坏情况下排序算法的计算复杂度的界,任何排序算法不能超越此界.

可以将这个问题归结为如下的问题:S 为含有 m 个元素的集合. A 从中任取一个元素,要求 B 提出 k 个"是"与"非"的问题后判断所取的元素. k 的下界是什么?也就是 B 得到 k 个问题"是"与"非"的回答后,任何情况下都能判断所取的元素.

假定第1个"是"、"非"问题后可将 S 分成两部分:$Y^{(1)}$ 与 $N^{(1)}$,假定 $|Y^{(1)}| \geqslant m/2$,而且假定所取的元素在 $Y^{(1)}$. B 提出的第2个问题又将 $Y^{(1)}$ 分成 $Y^{(2)}$ 和 $N^{(2)}$,假定 $|Y^{(2)}| \geqslant m/4$,且所取的元素在 $Y^{(2)}$,…,B 通过提问题将搜索的空间逐渐缩小,直至最后只剩下一个元素,即

$$\frac{m}{2^k} \leqslant 1, \quad k \geqslant \operatorname{lb} m.$$

n 个元素的排列数为 $n!$,排序可以看作是从 $n!$ 个可能中找到这个排列."是"与"非"问题可以说是排序算法中必不可少的"比较",故 $m = n!$ 根据斯特灵(Stirling)近似公式

$$n! \sim \sqrt{2n\pi}\left(\frac{n}{e}\right)^n.$$

$$\operatorname{lb}(n!) \approx n\operatorname{lb} n - \frac{1}{2}\operatorname{lb} n - n\operatorname{lb} e + O(1).$$

$O(1)$ 是常数. 故排序问题的下界为 $O(n\operatorname{lb} n)$. 也就是说排序算法的最坏情况下的复杂度不可能低于 $O(n\operatorname{lb} n)$.

7.2 归并排序算法

归并排序算法的基本思想是将待排序的序列

$$a_1, a_2, \cdots, a_n$$

一分为二:

$$a_1, a_2, \cdots, a_l,$$

$$a_{l+1}, a_{l+2}, \cdots, a_n, \quad l = \left\lceil \frac{n}{2} \right\rceil;$$

两个子序列分别进行排序,待各自排序完毕后进行归并. 假定 $a_1 < a_2 < \cdots < a_m$, $b_1 < b_2 < \cdots < b_n$,将二者归并为 $c_1 < c_2 < \cdots < c_{n+m}$. 归并算法如下:

步1 $k \leftarrow 1, j \leftarrow 1, i \leftarrow 1$.

步2 若 $i \leqslant m$ 且 $j \leqslant n$ 则转步3,否则转步6.

步3 若 $a_i < b_j$ 则转步4,否则作

【$c_k \leftarrow b_j, j \leftarrow j+1$,转步5】.

步4 $c_k \leftarrow a_i, i \leftarrow i+1$.

步5 $k \leftarrow k+1$,转步2.

步 6 若 $i>m$ 则转步 7,否则转步 9.

步 7 $c_k \leftarrow b_j, j \leftarrow j+1, k \leftarrow k+1$.

步 8 若 $j \leqslant n$ 则转步 7.否则结束.

步 9 $c_k \leftarrow a_i, i \leftarrow i+1, k \leftarrow k+1$.

步 10 若 $i \leqslant m$,则转步 9,否则结束.

归并排序算法应递归调用,即分成两个子序列后,对子序列的排序继续利用归并排序算法,直到解下一个元素为止.归并算法主要的工作在于归并,是分治策略的典型例子.

归并算法的复杂性分析:

(1) 最好情况.设 T_n 表 2^n 个元素用归并算法所作的比较次数,则

$$T_n = 2T_{n-1} + 2^{n-1}, \quad T_1 = 1,$$

即

$$T_n = \frac{1}{2} n 2^n.$$

(2) 最坏情况.

$$T_n = 2T_{n-1} + 2^n - 1, \quad T_1 = 1,$$

即

$$T_n = (n-1)2^n + 1.$$

(3) 空间复杂性分析:存储单元为 $2N$,N 为排序的元素数目.

7.3 快速排序算法

快速排序算法也是分治策略的典型例子.它的基本思想是将待排序的序列分成两部分,一部分比 k 小,另一部分大于 k,k 是序列中某一元素.对两个子序列继续利用快速排序算法进行排序.与归并排序不同的是,两个子序列排序完毕后无需进行归并,只要拼接起来就可以了.最简单的办法取开始的元素作为 k.例如

3 9 1 6 5 4 8 2 10 7

(1) 引进指针 i 和 j 分别指向序列的开始和末端的元素,例如

```
3  9  1  6  5  4  8  2  10  7
↑                           ↑
i                           j
```

(2) 指针 j 向左移,直到比 3 小的数停止,本例为 2,然后 2 与 3 互换位置得

```
2  9  1  6  5  4  8  3  10  7
↑                    ↑
i                    j
```

(3) 指针 i 向右移动,直到比 3 大的数 9 停住,9 与 3 互换位置得

```
2  3  1  6  5  4  8  9  10  7
   ↑                 ↑
   i                 j
```

(4) 指针 i 与 j 间继续上面 1—3 的步骤,直到 i 与 j 重合为止.如 j 左移到 1,得

2　3　1　6　5　4　8　9　10　7
　↑　↑
　i　j

1与3互换得

2　1　3　6　5　4　8　9　10　7
　　↑↑
　　ij

这时已将待排序的序列分成两个子序列：

2　1

6　5　4　8　9　10　7

继续对两个子序列采用快速排序.

为了避免最坏情况出现，可用随机的办法在待排序的序列中产生元素 k，以取代的序列的第一个元素为 k.

快速排序算法的复杂性分析：

(1) 最坏情况.即出现待排序的序列满足

$$a_n > a_{n-1} > a_{n-2} > \cdots > a_2 > a_1$$

或

$$a_1 > a_2 > \cdots > a_{n-1} > a_n,$$

这时若 T_n 代表 n 个元素快速排序法需要的比较次数，则

$$T_n = T_{n-1} + n - 1, \quad T_2 = 1,$$

即

$$T_n = \frac{1}{2}n(n-1),$$

(2) 最好情况.分成的两个子序列长度一样.设 $n = 2^k$，令 T_k 为 2^k 个元所作的比较次数，则

$$T_k = 2T_{k-1} + 2^k, \quad T_1 = 1.$$

令

$$G(x) = 1 + T_2 x + T_3 x^2 + \cdots,$$

$$\begin{array}{lll} x: & T_2 & = 2T_1 + 2^2 \\ x^2: & T_3 & = 2T_2 + 2^3 \\ +) & & \vdots \\ \hline & G(x) - 1 = & 2xG(x) + \dfrac{4x}{1-2x} \end{array}$$

$$(1-2x)G(x) = \frac{1+2x}{1-2x},$$

$$G(x) = \frac{1+2x}{(1-2x)^2} = (1+2x)\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{1}{2}\left(\frac{1}{1-2x}\right)\right]$$

$$= \frac{(1+2x)}{2}[1 + (2x) + (2x)^2 + \cdots]'$$

$$=\frac{1+2x}{2}[2+2\cdot 2(2x)+3\cdot 2\cdot(2x)^2+\cdots],$$

得
$$T_k=k\cdot 2^k+2(k-1)2^{k-1}=2k2^k-2^k.$$

代人
$$n=2^k,\quad k=\mathrm{lb}\,n.$$

所以最好情况的比较次数为 $2n\mathrm{lb}\,n-n$

(3) 平均情况. 令 T_n 表示 n 个元素利用快速排序算法平均的比较次数. 假定取第 k 个元素, 将序列分成两部分; 这个 k 从 $1,2,\cdots,n$ 机会均等. 所以

$$\overbrace{\underbrace{\quad}_{k-1}\ k\ \underbrace{\quad}_{n-k}}$$

$$T_n=\frac{1}{n}\sum_{n=1}^{n}(n-1+T_{k-1}+T_{n-k})=n-1+\frac{2}{n}\sum_{k=1}^{n-1}T_k,$$

$$T_0=0.$$

由
$$nT_n=n(n-1)+2\sum_{k=0}^{n-1}T_k,$$

$$(n+1)T_{n+1}=n(n+1)+2\sum_{k=0}^{n}T_k,$$

得

$$(n+1)T_{n+1}-nT_n=2n+2T_n,$$

$$(n+1)T_{n+1}=(n+2)T_n+2n.$$

令 $T_n=(n+1)s_n$, 则有

$$s_{n+1}-s_n=\frac{2n}{(n+1)(n+2)},\quad s_0=0,$$

即

$$s_1=0,$$

$$s_2=\frac{2\cdot 1}{2\cdot 3},$$

$$s_3=s_2+\frac{2\cdot 2}{3\cdot 4}=\frac{2\cdot 1}{2\cdot 3}+\frac{2\cdot 2}{3\cdot 4}.$$

令

$$s_{n-1}=\frac{2\cdot 1}{2\cdot 3}+\frac{2\cdot 2}{3\cdot 4}+\cdots+\frac{2(n-2)}{(n-1)n},$$

则

$$\begin{aligned}s_n&=\frac{2\cdot 1}{2\cdot 3}+\frac{2\cdot 2}{3\cdot 4}+\cdots+\frac{2(n-1)}{n(n+1)}\\&=\sum_{k=1}^{n-1}\frac{2k}{(k+1)(k+2)}=4\sum_{k=1}^{n-1}\frac{1}{k+2}-2\sum_{k=1}^{n-1}\frac{1}{k+1}\\&=4\sum_{k=2}^{n}\frac{1}{k+1}-2\sum_{k=1}^{n-1}\frac{1}{k+1}=2\sum_{k=2}^{n}\frac{1}{k+1}+\frac{4}{n+1}-1.\end{aligned}$$

但

$$\sum_{k=2}^{n-1} \frac{1}{k+1} = \sum_{k=3}^{n} \frac{1}{k} < \int_2^n \frac{1}{x} \mathrm{d}x = \ln n - \ln 2.$$

所以

$$s_n < 2\ln n - \left(2\ln 2 - \frac{4}{n+1} + 1\right),$$

$$s_n < 2\ln n + c,$$

$$T_n = (2n+2)\ln n + \Theta(n),$$

其中 $\Theta(n)$ 为 n 一次方项函数.

7.4 堆集排序算法

前面介绍的排序算法是基于分治策略.这一节讨论的堆集排序既是基于"优先策略"同时还借助于一种叫做堆的数据结构.

(1) 堆是一种二分树,例如图 7-1 所示.

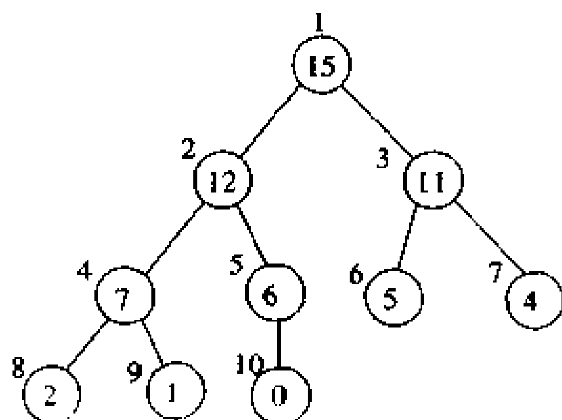


图 7-1

图中每一节点有两个数:一是○里的数,一是○外面的数;前者是该点存储的数,后者是堆的地址序号.假如一节点的地址为 n,则它的父亲节点的地址为 $\lceil \frac{n}{2} \rceil$,它的左儿子节点的地址为 $2n$,右儿子节点的地址为 $2n+1$.它们的地址关系可用图 7-2 表示.

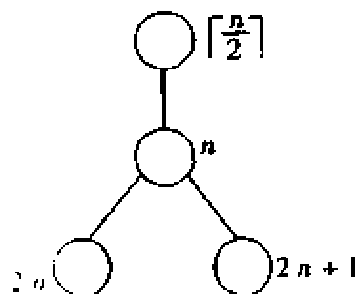


图 7-2

$\lceil \frac{n}{2} \rceil$,$2n$,$2n+1$ 用 2 进制表示时,运算十分方便.

堆除了地址有如上特点的二分树以外,还要求每一节点所储存的数比它的儿子节点的数要大.因此堆的根节点储存的数必是储存在堆里的数的最大者.

(2) **建堆算法**.用例子叙述建堆方法,如图 7-3.

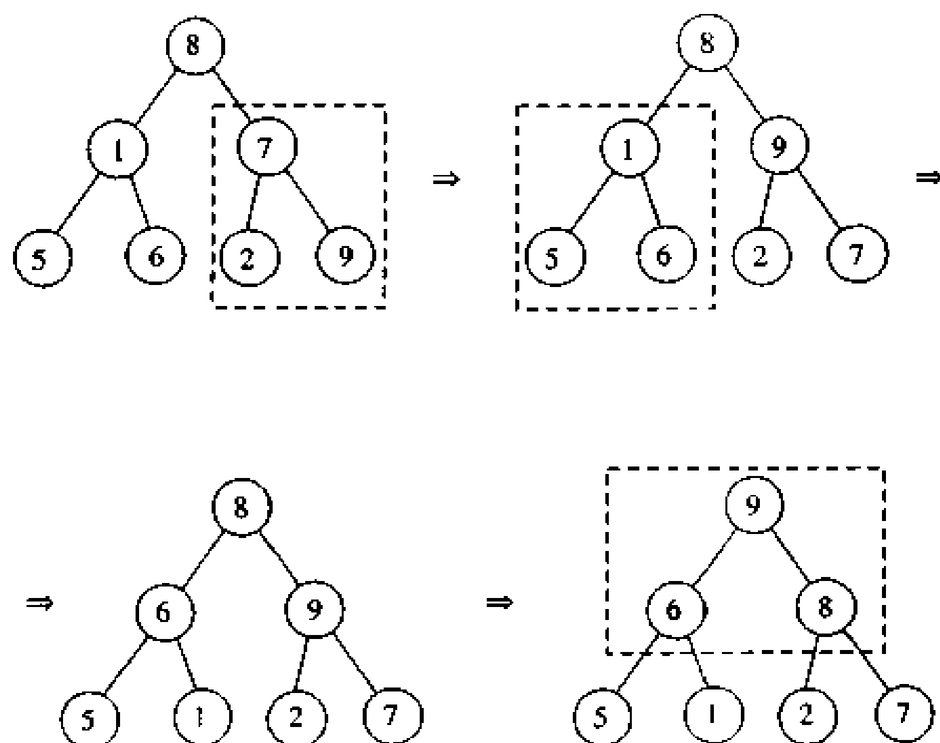


图 7-3

图中虚线框表示考察的父子一组三个节点，检查是否满足堆的要求：父亲节点存储的数比左、右两儿子节点存储的数大.

建堆的复杂性分析：

堆顶有 1 个顶点；

第 1 层有 2 个顶点；

第 2 层有 2^2 个顶点；

……

建堆过程假定各内点要调整到满足堆的要求为止.

设堆的高度为 h，顶点个数

$$n = 1 + 2 + 2^2 + \cdots + 2^h = 2^{h+1} - 1.$$

第 k 层共 2^k 个顶点，该层的点最坏情况要“下沉”到叶子节点为止，必须做 $2(h-k)$次比较．故建堆过程最坏情况要作比较次数为

$$s = 2\sum_{k=0}^{h}(h-k)2^k = 2h\sum_{k=0}^{h}2^k - 2\sum_{k=0}^{h}k2^k,$$

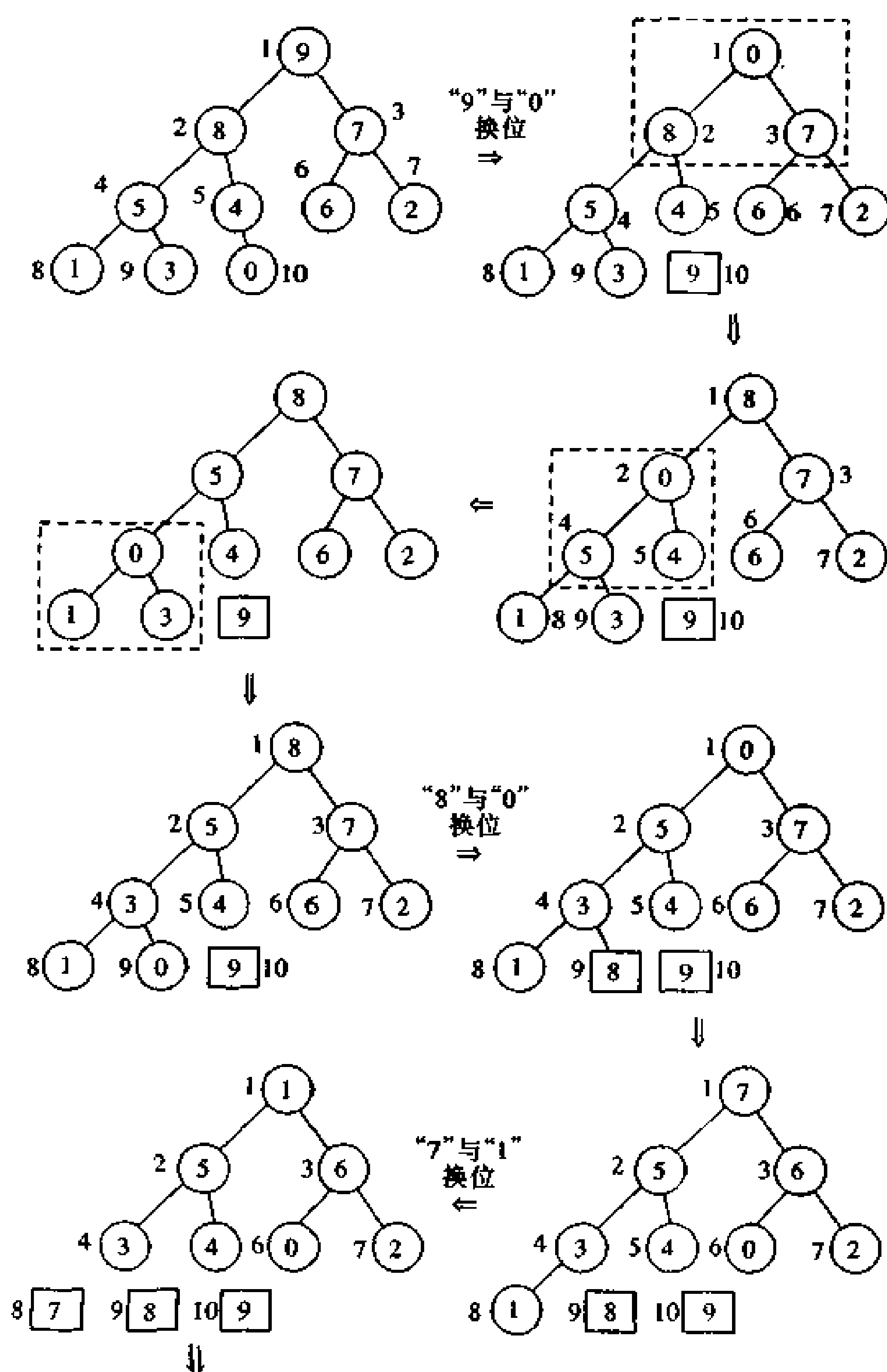
$$\sum_{k=0}^{h}k2^k = (x + 2x^2 + \cdots + kx^k)_{x=2} = [x(1 + 2x + \cdots + kx^{k-1})]_{x=2}$$

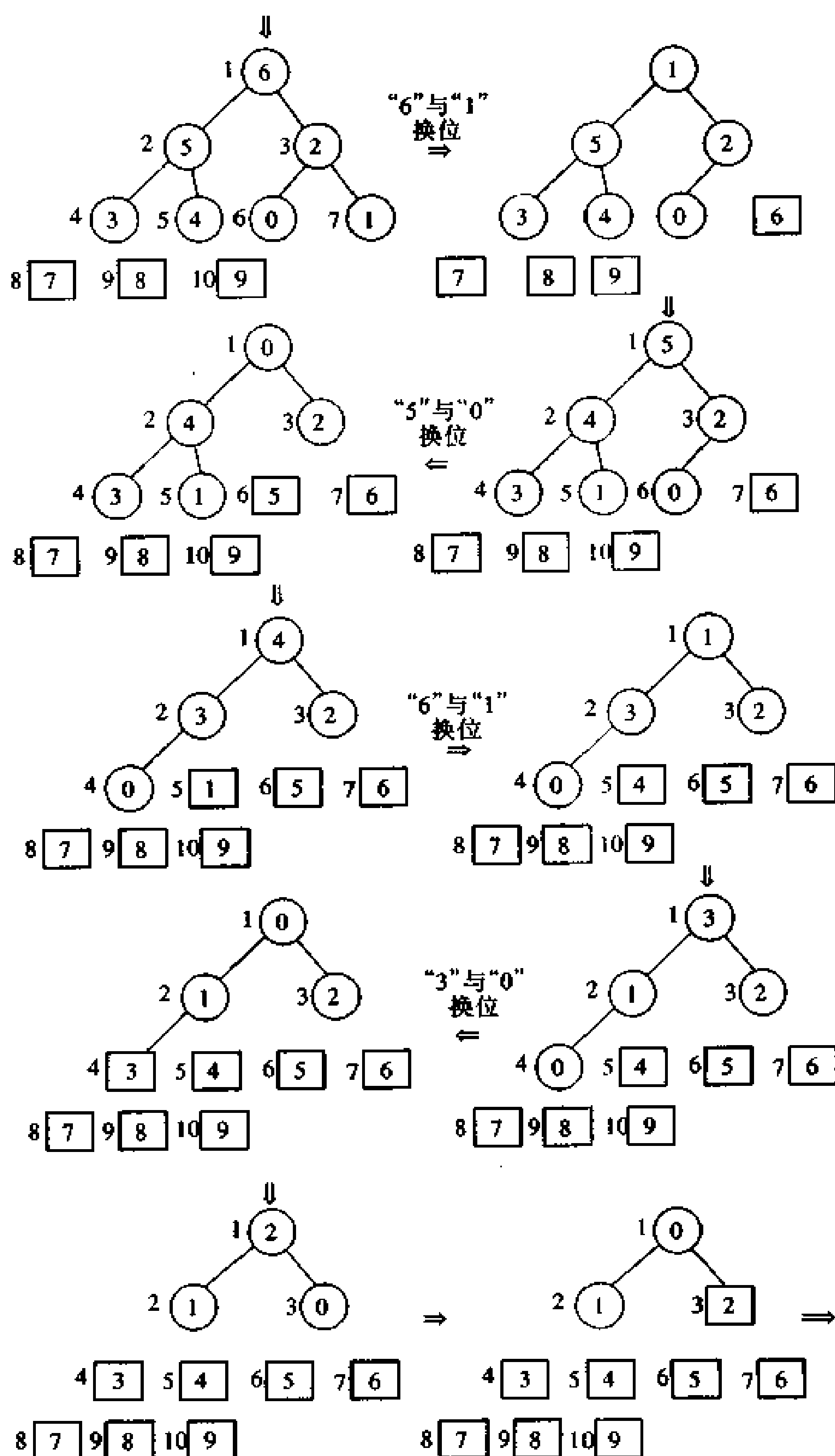
所以

$$s = 2(2^{h+1} - h) - 4 \approx 2n.$$

(3) **堆集排序法**. 假定已经建成堆，堆顶元素为最大值，将堆顶元素与最右下方元素换位. 换位后进行调整使之恢复成为堆. 如此循环反复直到最后.

举例如图 7-4.





"6"与"1"
换位
"5"与"0"
换位
"6"与"1"
换位
"3"与"0"
换位

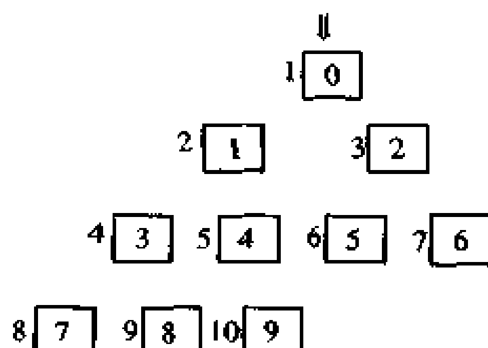


图 7-4

k 个节点的堆高为[lb$k$].堆集排序法经常要将底层的元素置于堆顶,恢复成堆将做 2[lb$k$]次比较.故堆集排序要作的比较次数为

$$s = 2\{([\mathrm{lb}2] + [\mathrm{lb}3]) + ([\mathrm{lb}4] + [\mathrm{lb}5] + [\mathrm{lb}6] + [\mathrm{lb}7]) + \cdots + (\cdots + [\mathrm{lb}n])\}$$

$$\leqslant 2\{2\cdot 1 + 2\cdot 2^2 + 3\cdot 2^3 + \cdots + [\mathrm{lb}h]2^{[\mathrm{lb}n]}\}.$$

堆集排序算法的时间复杂性为 $\Theta(n\mathrm{lb}n)$.

堆是一种很重要的数据结构,堆集排序是十分出色的排序算法.堆还有一种非常重要的应用,如计算机任务的安排,按优先级安排其加工顺序,虽然对象随时有新的加入或删除,但只要将优先级最高的元素置于堆顶,无需排序.当最先的任务取走后或加入新任务或删除某任务时,只要适当调整永远保持是堆,优先级最高的任务置于堆顶就可以了.

7.5 排序网络

这一节将介绍排序的并列计算装置——排序网络.排序网络是由比较器和导线组成,如图 7-5 所示.

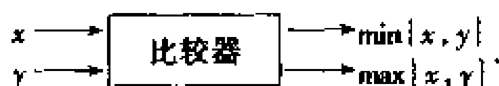


图 7-5

以后为方便起见,习惯用竖线表示它,如图 7-6.

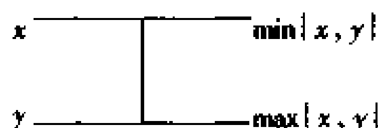


图 7-6

巴特赛尔(Batcher)奇偶归并网络:

假定 $a_1 < a_2 < \cdots < a_m, b_1 < b_2 < \cdots < b_n$ 是两组已排序的序列,巴特赛尔奇偶归并算法如下:

步 1　$a_1, a_3, a_5\cdots$ 和 $b_1, b_3, b_5, \cdots$ 归并得 $c_1, c_2, c_3, \cdots$; $a_2, a_4, a_6, \cdots$ 和 $b_2, b_4,$

$b_6,\cdots$归并得 $d_1,d_2,d_3,\cdots$.

步2 对于序列 $c_1,d_1,c_2,d_2,\cdots$,置比较元件于(2,3),(4,5),…间.这里(2,3),(4,5)即用比较元件联接于 d_1 和 c_2,d_2 和 c_3 之间.余此类推.

例1 $m=2,n=1$ 时的巴特赛尔归并网络如图7-7所示,用 $B(2,1)$表示它.请注意 $B(2,1)$指的是虚线包围的部分.下同此.

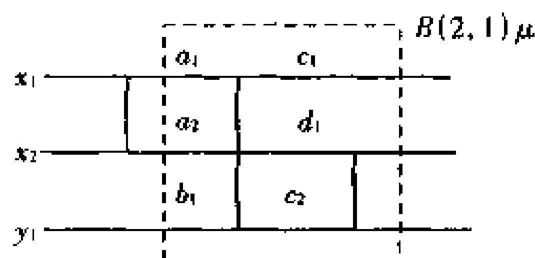


图7-7

图7-8是 $B(2,2)$网络,图7-9是 $B(3,2)$网络.

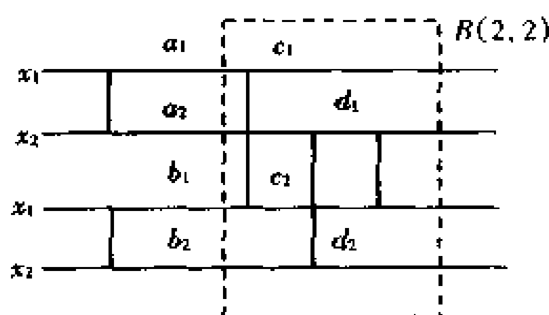


图7-8

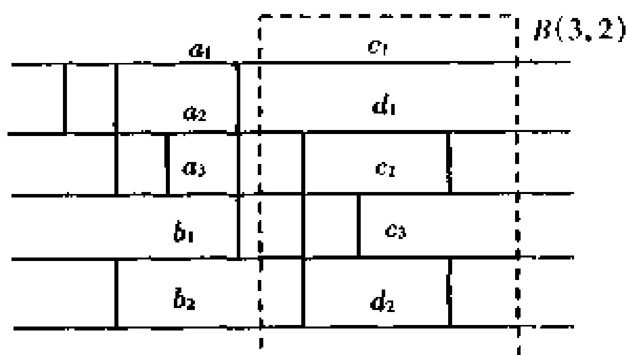


图7-9

最后介绍0-1原理,用以检验网络是否工作正确.对于有 n 个输入端的排序网络(图7-10),要验证它的工作正确,势必要对 $x_1,x_2,\cdots,x_n$ 的 $n!$ 个排列一一验证,对所有的状态输入结果都正确,方可确认该排序网络正确.0-1原理是说只要对于输入端 $x_1,x_2,\cdots,x_n$ 作{0,1}序列验证即可,即只要对 2^n 种状态输入,结果正确便可确认该排序网络是正确的.

图7-10

对于比较元件,若 $f(x)$是单调增函数,则有如图7-11的输出.

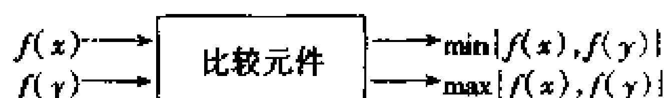


图 7-11

对于排序网络，当输入为 $a_1,a_2,\cdots,a_n$ 时，有输出依次为 $b_1,b_2,\cdots,b_n$，若 $f(x)$ 是单调增函数，则当输入为 $f(a_1),f(a_2),\cdots,f(a_n)$ 时，输出依次为 $f(b_1),f(b_2),\cdots,f(b_n)$，即如图 7-12 所示.

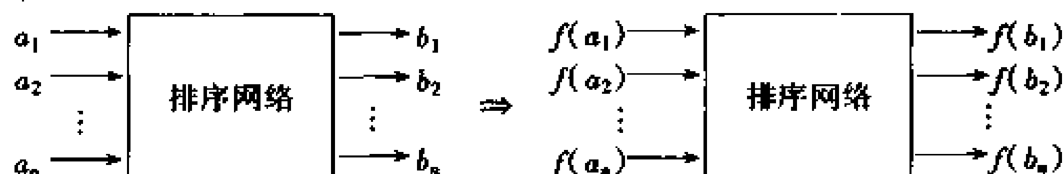


图 7-12

假定对所有的 0-1 序列工作正确的排序网络，存在一个反例，即假定一输入为 $\{a_1,a_2,\cdots,a_n\}$，输出的结果对其中的 $a_i<a_j$，a_j 的输出置于了 a_i 输出的前面. 定义一单调增函数 $f(x)$，

$$f(x)=\begin{cases}0, & x<a_i,\\ 1, & x\geqslant a_i,\end{cases}$$

也就是说 $f(a_j)=1$. 这就与对 0-1 序列工作正确的假定矛盾. 这就证明了对于任意排序网络工作不正确，必然对于 0-1 序列工作不正确. 对 0-1 序列工作不正确是排序网络工作不正确的必要条件.

7.6　最佳二分树

前面各节讨论的是排序，后面转入讨论查找. 假定有 n 个文件储存在计算机系统内，已知它们的关键字 $a_1<a_2<\cdots<a_n$，它们的查找概率分别为 $p_1,p_2,\cdots,p_n$. 又 $z_0,z_1,\cdots,z_n$ 为查找失败的 $n+1$ 种状态：

$$z_0<a_1,a_1<z_1<a_2,\cdots,$$
$$a_{n-1}<z_{n-1}<a_n,a_n<z_n.$$

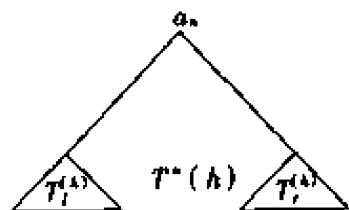


图 7-13

设 q_j 是 z_j 事件发生的概率，$j=0,1,2,\cdots,n$.

$$(p_1+p_2+\cdots+p_n)+(q_0+q_1+\cdots+q_n)=1.$$

假定 n 个文件的存储策略是一二分树 T，二分树的每一条边的长度设为 1，a_i 点到根节点的距离为 l_i，则该二分树的平均查找次数为

$$m(T)=\sum_{i=1}^{n}p_i(l_i+1)+\sum_{j=0}^{n}q_j\bar{l}_j,$$

其中 $\bar{l}_j$ 为叶子 z_j 到根节点的距离. 平均查找次数最少的二分树称为**最佳二分树**，假

定该最佳二分树以 a_h 为根节点，用 $T^*(h)$ 表示它，它的左、右子二分树为 $T_l^{(h)}$，$T_r^{(h)}$ 见图 7-13. $T_l^{(h)}$ 是包含 $a_1,a_2,\cdots,a_{h-1}$ 及 $z_0,z_1,\cdots,z_{h-1}$ 的左子二分树，$T_r^{(h)}$ 是包含 $a_{h+1},a_{h+2},\cdots,a_n$ 及 $z_h,z_{h+1},\cdots,z_n$ 的右子二分树. 令

$$m(T^*(h)) = \sum_{i=1}^{n} p_i(l_i+1) + \sum_{j=0}^{n} q_j \bar{l}_j$$
$$= m(T_l^{(h)}) + m(T_r^{(h)}) + p_h + l_h + r_h,$$

其中

$$l_h = \sum_{i=1}^{h-1} p_i + \sum_{j=0}^{h-1} q_j, \quad r_h = \sum_{i=h+1}^{n} p_i + \sum_{j=h}^{n} q_j,$$
$$p_h + l_h + r_h = p_1 + p_2 + \cdots + p_n + q_0 + q_1 + \cdots + q_n.$$

若 $T^*(h)$ 是以 a_h 为根节点的最佳二分树，则 $T_l^{(h)}$ 和 $T_r^{(h)}$ 也一定是最佳二分树.

$$m(T^*(h)) \xlongequal{\text{def}} \min_k \{m(T(k))\}$$
$$= \min_k \{m(T_l^{*(k)}) + m(T_r^{*(k)})\} + \sum_{i=1}^{n} p_i + \sum_{j=0}^{n} q_j.$$

可以把以上的讨论推至包含 $a_i,a_{i+1},\cdots,a_j$ 及 $z_{i-1},z_i,\cdots,z_j$ 的任一子二分树. 这样的子二分树用 $T\begin{pmatrix} h \\ i,j \end{pmatrix}$ 表示它，其中 h、i、j 为该子二分树的节点的下标，即包含内点 $a_i,a_{i+1},\cdots,a_j$ 及叶子 $z_{i-1},z_i,\cdots,z_j$，以 a_h 为根节点的二分树，其中 $i \leqslant h \leqslant j$. $T^*(i,j)$ 表包含 $a_i,a_{i+1},\cdots,a_j$ 及 $z_{i-1},z_i,\cdots,z_j$ 的最佳二分树.

用动态规划求最佳二分树的方法举例说明如下：已知 $a_1<a_2<a_3<a_4$，$p_1=p_2=3/16$，$p_3=2/16$，$p_4=1/16$. $q_0=q_1=2/16$，$q_2=q_3=q_4=1/16$.

为计算方便起见，将分母 16 略去，仅就分子进行计算.

(1) 有一个内点的情况：

$$m\left(T\begin{pmatrix} 1 \\ 1,1 \end{pmatrix}\right) = m\left(\begin{array}{ccc} & a_1 & \\ \swarrow & & \searrow \\ z_0 & & z_1 \end{array}\right) = p_1+q_0+q_1 = 3+2+2 = 7,$$

$$m\left(T\begin{pmatrix} 2 \\ 2,2 \end{pmatrix}\right) = m\left(\begin{array}{ccc} & a_2 & \\ \swarrow & & \searrow \\ z_1 & & z_2 \end{array}\right) = p_2+q_1+q_2 = 3+2+1 = 6,$$

$$m\left(T\begin{pmatrix} 3 \\ 3,3 \end{pmatrix}\right) = m\left(\begin{array}{ccc} & a_3 & \\ \swarrow & & \searrow \\ z_2 & & z_3 \end{array}\right) = p_3+q_2+q_3 = 2+1+1 = 4,$$

$$m\left(T\begin{pmatrix} 4 \\ 4,4 \end{pmatrix}\right) = m\left(\begin{array}{ccc} & a_4 & \\ \swarrow & & \searrow \\ z_3 & & z_4 \end{array}\right) = p_4+q_3+q_4 = 1+1+1 = 3.$$

(2) 有两个内点的情况：

$$m(T^*(1,2)) = \min\left\{ m\left(T\begin{pmatrix} 1 \\ 1,2 \end{pmatrix}\right), m\left(T\begin{pmatrix} 2 \\ 1,2 \end{pmatrix}\right)\right\}$$

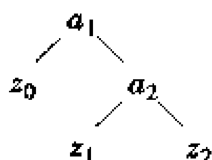
$$= \min\left\{ m\left(\begin{array}{ccccc} & a_1 & & & \\ z_0 & & a_2 & & \\ & z_1 & & z_2 & \end{array}\right), m\left(\begin{array}{ccccc} & & a_2 & & \\ & a_1 & & z_2 & \\ z_0 & & z_1 & & \end{array}\right)\right\}$$

$$= \min\left\{ m\left(\begin{array}{ccc} & a_2 & \\ z_1 & & z_2 \end{array}\right), m\left(\begin{array}{ccc} & a_1 & \\ z_0 & & z_1 \end{array}\right)\right\} + p_1 + p_2 + q_0 + q_1 + q_2$$

$$= \min\{\underline{6}, 7\} + 6 + 5 = 17,$$

所以
$$T^*(1,2) = T\binom{1}{1,2},$$

$T^*(1,2)$:



$$m(T^*(2,3)) = \min\left\{ m\left(T\binom{2}{2,3}\right), m\left(T\binom{3}{2,3}\right)\right\}$$

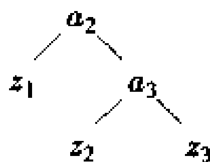
$$= \min\left\{ m\left(\begin{array}{ccccc} & a_2 & & & \\ z_1 & & a_3 & & \\ & z_2 & & z_3 & \end{array}\right), m\left(\begin{array}{ccccc} & & a_3 & & \\ & a_2 & & z_3 & \\ z_1 & & z_2 & & \end{array}\right)\right\}$$

$$= \min\left\{ m\left(\begin{array}{ccc} & a_3 & \\ z_2 & & z_3 \end{array}\right), m\left(\begin{array}{ccc} & a_2 & \\ z_1 & & z_2 \end{array}\right)\right\} + p_2 + p_3 + q_1 + q_2 + q_3$$

$$= \min\{\underline{4}, 6\} + 9 = 13,$$

所以
$$T^*(2,3) = T\binom{2}{2,3},$$

$T^*(2,3)$:



$$m(T^*(3,4)) = \min\left\{ m\left(T\binom{3}{3,4}\right), m\left(T\binom{4}{3,4}\right)\right\}$$

$$= \min\left\{ m\left(\begin{array}{ccccc} & a_3 & & & \\ z_2 & & a_4 & & \\ & z_3 & & z_4 & \end{array}\right), m\left(\begin{array}{ccccc} & & a_4 & & \\ & a_3 & & z_4 & \\ z_2 & & z_3 & & \end{array}\right)\right\}$$

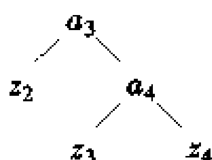
$$= \min\left\{ m\left(\begin{array}{ccc} & a_4 & \\ z_3 & & z_4 \end{array}\right), m\left(\begin{array}{ccc} & a_3 & \\ z_2 & & z_3 \end{array}\right)\right\} + p_3 + p_4 + q_2 + q_3 + q_4$$

$$= \min\{\underline{3}, 4\} + 3 + 3 = 9,$$

所以

$$T^*(3,4)=T\binom{3}{3,4},$$

$T^*(3,4)$:



(3) 有三个内点的情况：

$$m(T^*(1,3))=\min\left\{m\left(T\binom{1}{1,3}\right),m\left(T\binom{2}{1,3}\right),m\left(T\binom{3}{1,3}\right)\right\}$$

$$=\min\left\{m\left(\begin{array}{ccccc} & a_1 & & & \\ z_0 & & a_2 & & \\ & z_1 & & a_3 & \\ & & z_2 & & z_3\end{array}\right),m\left(\begin{array}{ccc} & a_2 & \\ a_1 & & a_3 \\ z_0\ \ z_1 & & z_2\ \ z_3\end{array}\right),m\left(\begin{array}{ccccc} & & & a_3 & \\ & a_1 & & & z_3 \\ z_0 & & a_2 & & \\ & z_1 & & z_2 & \end{array}\right)\right\}$$

$$=\min\left\{m\left(\begin{array}{cccc} & a_2 & & \\ z_1 & & a_3 & \\ & z_2 & & z_3\end{array}\right),m\left(\begin{array}{ccc} & a_1 & \\ z_0 & & z_1\end{array}\right)+m\left(\begin{array}{ccc} & a_3 & \\ z_2 & & z_3\end{array}\right),\right.$$

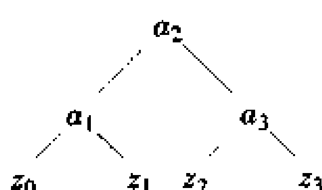
$$\left.m\left(\begin{array}{cccc} & a_1 & & \\ z_0 & & a_2 & \\ & z_1 & & z_1\end{array}\right)\right\}+p_1+p_2+p_3+q_0+q_1+q_2+q_3$$

$$=\min\{13,\underline{11},17\}+8+6=25,$$

所以

$$T^*(1,3)=T\binom{2}{1,3},$$

$T^*(1,3)$:



$$m(T^*(2,4))=\min\left\{m\left(T\binom{2}{2,4}\right),m\left(T\binom{3}{2,4}\right),m\left(T\binom{4}{2,4}\right)\right\}$$

$$=\min\left\{m\left(\begin{array}{ccccc} & a_2 & & & \\ z_1 & & a_3 & & \\ & z_2 & & a_4 & \\ & & z_3 & & z_4\end{array}\right),m\left(\begin{array}{ccc} & a_3 & \\ a_2 & & a_4 \\ z_1\ \ z_2 & & z_3\ \ z_4\end{array}\right),\right.$$

$$
m\left(\begin{array}{ccccc} & & & a_4 & \\ & & \diagup & & \diagdown \\ & a_2 & & & z_4 \\ \diagup & & \diagdown & & \\ z_1 & & a_3 & & \\ & \diagup & & \diagdown & \\ z_2 & & & & z_3 \end{array}\right)\Bigg\}
$$

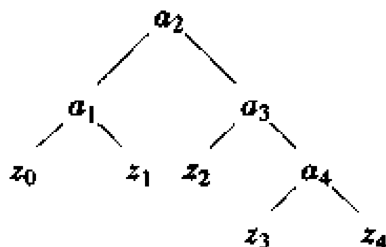
$$
\begin{aligned}
&= \min\{m(T^*(3,4)),\ m(T^*(2,2)) + m(T^*(4,4)),\ m(T^*(2,3))\} + \\
&\quad p_2 + p_3 + p_4 + q_1 + q_2 + q_3 + q_4 \\
&= \min\{\underline{9}, \underline{9}, 13\} + 11 = 20,
\end{aligned}
$$

所以
$$T^*(2,4) = T\binom{2}{2,4} = T\binom{3}{2,4},$$

(4) 有四个内点情况:

$$
\begin{aligned}
m(T^*(1,4)) = \min\Bigg\{ & m\left(\begin{array}{ccc} & a_1 & \\ \diagup & & \diagdown \\ z_0 & & T^*(2,4) \end{array}\right),\ m\left(\begin{array}{ccc} & a_2 & \\ \diagup & & \diagdown \\ T^*(1,1) & & T^*(3,4) \end{array}\right), \\
& m\left(\begin{array}{ccc} & a_3 & \\ \diagup & & \diagdown \\ T^*(1,2) & & T(4,4) \end{array}\right),\ m\left(\begin{array}{ccc} & a_4 & \\ \diagup & & \diagdown \\ T^*(1,3) & & z_4 \end{array}\right)\Bigg\} \\
= \min\{ & m(T^*(2,4)),\ m(T(1,1)) + m(T^*(3,4)), \\
& m(T^*(1,2)) + m(T(4,4)),\ m(T^*(1,3))\} + 16 \\
= \min\{ & 20, \underline{7+9}, 17+3, 25\} + 16 = 32,
\end{aligned}
$$

所以
$$T^*(1,4) = T\binom{2}{1,4},$$



7.7　2-3 树

文件存储于计算机系统内,供查询用.一般要求储存的树的高度比较均衡,这时查询所作的比较次数的平均值最少.但是文件不是固定不变的,有时要插入新

的,有时还要删除某些文件,本来比较均衡的树经过一番插入和删除后,性态恶化.下面介绍一种数据结构,遇到插入或删去后容易调整,使之较快地恢复到比较良好的状态.2-3 树是其中典型的一种.

2-3 树是一棵树状结构,每一内容有两个值 a 和 b.它可能给出 3 个分支,如图 7-14 所示.

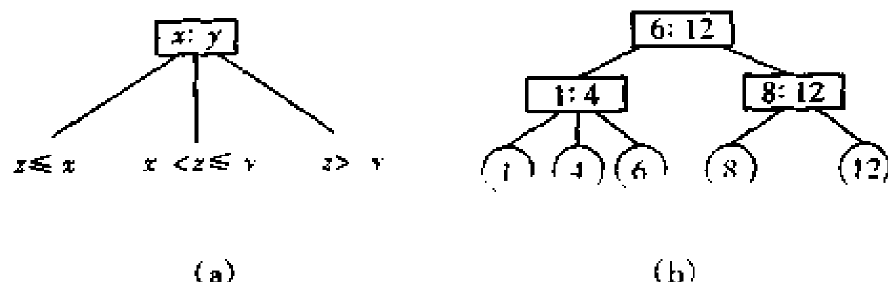


图 7-14

图 7-15 说明插入后如何调整使之保持 2-3 树的结构.

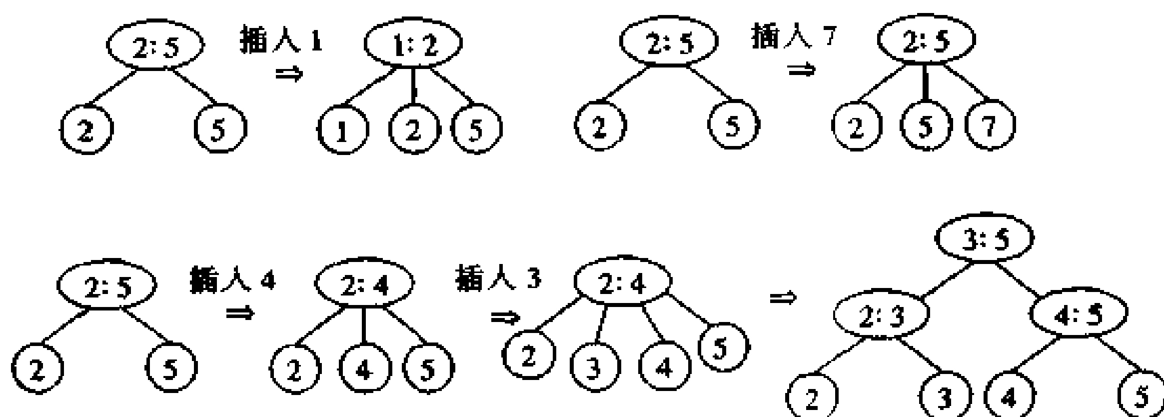


图 7-15

8 计算复杂性理论

8.1 概　　述

快速电子计算机贵在神速,它的出现使许多难题得到了解决,它可以对天气作出预报,可以操纵工厂的生产过程,甚至于驾驭战争.但它并非万能的!哪些问题可以通过它解决,哪些问题不可以通过它来解决?这属于可计算性理论研究的内容.理论上确定为可计算的问题,实际上未必可能,计算复杂性理论仅仅对那些理论上是可计算的问题,讨论它们实际上是否可计算.

一个问题可解,是指可设计一个算法,在有限步骤内给出答案.但这并不意味着该算法是有效的.如若一个算法需要计算机运行十个世纪才能结束,这算法实际上是不可行的.例如前面提到的汉诺塔问题便是一例.

当考虑一个算法的时间复杂度时,往往将它所需要运算的步骤看作是问题规模 n 的函数 $f(n)$. 计算机科学家有一个共识,当 $f(n)$ 是 n 的多项式界时,被认为是"好"算法,即是有效的. 本章里将介绍 NP 理论, NP 是 Nondeterministic Polynomial 的缩写. 它由库克(S. Cook)于 1971 年发表的"The Complexity of Theorem Proving Procedures"和卡普(R. Karp)于 1972 年发表的"Reducibility Among Comlinatorial Prolem"两篇论文奠定了基础.

8.2 图 灵 机

为了精确地刻画复杂度概念,引进图灵机概念.

设 $\{0,1\}^*$ 表示由 0、1 构成的有限长符号串的集合. $L \subseteq \{0,1\}^*$ 为它的一个子集,关于 L 的判定问题是指,任给 $x \in \{0,1\}^*$,判断 x 是否属于 L. 所有计算问题都可以归结为判定问题.

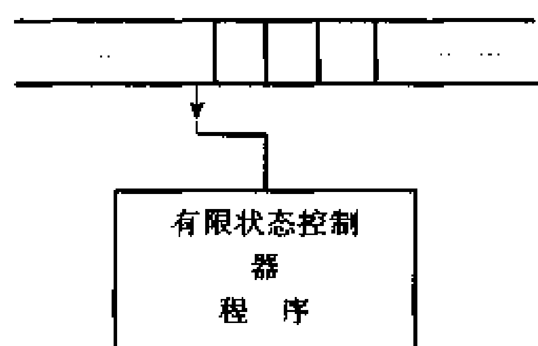


图 8-1

非形式地讲,图灵机由以下几部分组成:一条无限长的带,一个读写头,一个有限状态控制器及一段程序. 对于某一输入字符串,有限状态控制器通过读写头来读内容,当前状态决定动作,直到进入终结状态.

定义1 确定型图灵机 $\mathrm{TM} = \{Q, A, f\}$. 其中 Q 为有限状态集, q_0 为初始状态, q_Y 和 q_N 分别为接受和拒绝的终结状态, q_0、q_Y、q_N 是 Q 中三个特殊状态; $A = \{0, 1, \#\}$, 其中"#"为空格; f 为转移功能,它定义为

$$f: (Q \setminus \{q_Y, q_N\}) \times A \longrightarrow Q \times A \times \{r, l, h\},$$

r、l、h, 分别表示读写头右移、左移和不动. 转移功能表示图灵机在状态 $q \in Q$ 时,读写头读进字符 $a \in A$,则状态 q 改变为某一 $q' \in Q$,读写头在带上写上字符 $a' \in A$,然后读写头或右移一位或左移一位或不动.

图灵机的计算是通过图像(或格局)转移来进行的,所谓图像是指图灵机某一时刻的形象,由三元组 (q, w, i) 来表示,其中 q 为当前状态, w 为一字符串,它表示带上的情况, i 为当前读写头位置. 图像的转移由转移功能来控制,若有 $f(q, A_i) = (P, A, l)$,即当读写头读进 A_i 时,状态从 q 改为 p,而且读写头写上 A 并左移一位,则图像 $(q, A_1A_2\cdots A_i\cdots A_n, i)$ 转移到图像 $(p, A_1A_2\cdots A_{i-1}AA_{i+1}\cdots A_n, i-1)$. 若 $f(q, A_i) = (p, A, r)$ 或 $f(q, A_i) = (p, A, h)$,则图像 $(q, A_1A_2\cdots A_i\cdots A_n, i)$ 将分别改为 $(p, A_1A_2\cdots A\cdots A_n, i+1)$ 或 $(q, A_1A_2\cdots A\cdots A_n, i)$. 对于某一输入 w, TM 有着初始图像 $(q_0, w, 1)$,通过图像的转移,到达某一终态,则图灵机就完成了这次计算.

例如下述图灵机:

$$Q = \{q_0, q_1, q_Y, q_N\},$$

$$A = \{0,1,\#\},$$

f: 当前状态	读进字符	下一状态	写上字符	读写头位移
q_0	0	q_1	0	r
q_0	1	q_1	#	l
q_0	#	q_Y	#	h
q_1	0	q_0	1	r
q_1	1	q_1	0	r
q_1	#	q_r	#	h

则当输入 $w = 010100$ 时,运行过程如图 8-2 所示.

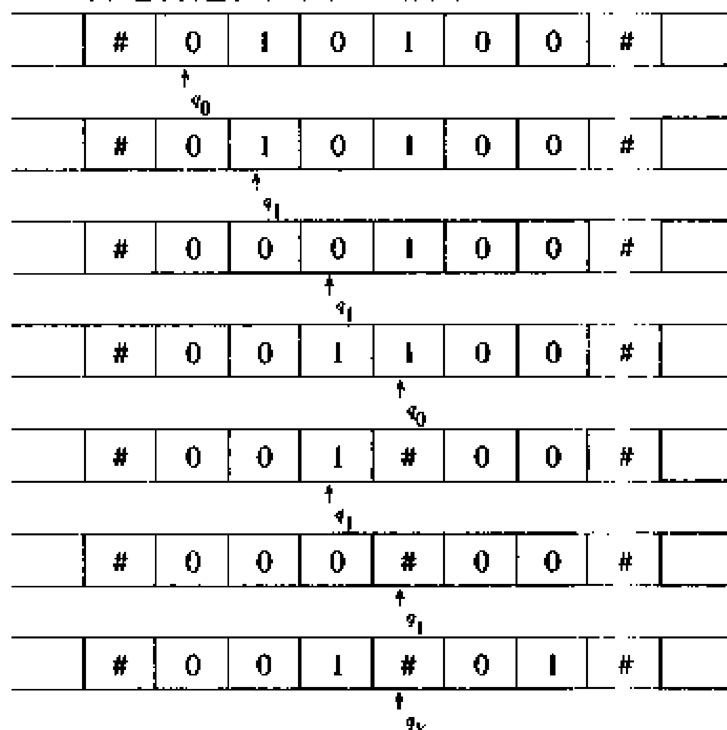


图 8-2

此时图灵机进入接受态 q_Y. 事实上,该图灵机接受的集合为

$$L = \{W \mid W \in \{0,1\}^*\}.$$

对于确定型图灵机,输入 x 后将会有三种可能:

(1) 图灵机进入状态 q_Y,停机并接受 x,

(2) 图灵机进入状态 q_N,停机并拒绝 x,

(3) 图灵机永不进入 q_Y 或 q_N,图灵机永不停机.

对于一个图灵机 TM,它接受的集合(语言)定义为

$$L_{TM} = \{x \mid \text{TM 对于输入 } x \text{ 将会停机并接受(即进入终态 } q_Y)\}.$$

集合 L 称作能为图灵机 TM 所识别,当且仅当若 $x \in L$,则 TM 停机并接受 x,否则停机并拒绝 x.

事实上,一个算法就是一个图灵机.若说一个算法解决了关于 L 的判定问题,相当于 L 能被该图灵机所识别.

有了图灵机,就可以定义算法的复杂度了.图灵机的每一次图像转移可看作是

完成一步计算，从输入(初始状态)开始到达终结状态的计算步数，为该图灵机对于输入 x 计算的时间复杂度.

设 $T(n)$ 为一个 $\mathbf{Z}^+ \to \mathbf{Z}^+$ 的函数，若对于任意长度 n 的输入 x，某一确定型的图灵机从初始状态 q_0 开始，至多在 $T(n)$ 步内终止，则称该图灵机在 $T(n)$ 时间内运行，或称该图灵机的计算(时间)复杂度为 $O(T(n))$.

定义多项式时间复杂类 P 如下：

$$\mathrm{P} \xlongequal{\text{def}} \left\{ L \subseteq \{0,1\}^* \;\middle|\; \begin{array}{l}\text{存在常数 } C \text{ 及时间复杂度为 } O(n^C) \\ \text{的图灵机 TM, TM 识别集合 } L\end{array} \right\}.$$

显然，P 就是以前所说的能在多项式时间内予以解决的问题的总和，也就是被称作能够有效解决的问题的集合.

以上介绍的是确定型图灵机，也就是说，在当前状态及符号确定的情况下，图灵机的下一图像(包括状态、带上符号及读写头位置)都是确定的，这样，每一输入 x 唯一地决定了计算的终结状态.

下面介绍非确定型图灵机，非确定型图灵机同样具有状态集 Q 及符号集 A，它与确定型图灵机的不同之处在于其状态转移功能为

$$f:(Q \setminus \{q_Y, q_N\}) \times A \longrightarrow Q \times A \times \{r, 1, h\} \text{的子集},$$

即它的图像转移是多值的，而非唯一确定的. 这样对于某一输入 x，它可能有多种计算路径而到达不同的终结状态. 称一个输入符号串 x 能为某一非确定性图灵机 NDTM 所接受，当且仅当该 NDTM 对于输入 x，存在着一条计算路径到达终态 q_Y，即停机接受态. 而为该 NDTM 所接受的集合为

$$L_{\mathrm{NDTM}} = \{x \mid \text{该 NDTM 接受 } x\}.$$

同样，可定义非确定型图灵机的时间复杂度. 设某一 NDTM 所接受的集合为 L，若有 $\mathbf{Z}^+ \to \mathbf{Z}^+$ 的函数 $T(n)$，对任意长度 n 的输入串 x，若 $x \in L$，则该 NDTM 存在一条计算路径接受 x，且该计算路径的计算步数(图像转移次数)至多为 $T(n)$ 步，则称该 NDTM 在 $T(n)$ 步内接受 x，或称该 NDTM 的时间复杂度为 $T(n)$. 类似于 P 的定义，可定义 NP 类如下：

$$\mathrm{NP} \xlongequal{\text{def}} \left\{ L \subseteq \{0,1\}^* \;\middle|\; \begin{array}{l}\text{存在一非确定型图灵机 NDTM 接受 } L, \\ \text{该 NDTM 的时间复杂度为输入长度 } n \text{ 的多项式}\end{array} \right\}.$$

直观地看，NP 类可看作可在多项式时间验证的问题的集合. 因为在非确定型图灵机中只要有一条计算路径接受 x，则认为该非确定图灵机接受 x，因而可将非确定型图灵机看作由许多处理器组成的并行计算装置，可同时进行计算. 也可以理解为它由两部分组成：一部分为猜测模块，它猜测能导致接受的那条计算路径，而另一部分为验证模块，它接收由猜测模块给出的计算路径，再依该路径来模拟原 NDTM 的行为. 由于计算路径已给出，故此时的验证为确定性的. 这样若 $x \in L_{\mathrm{NDTM}}$，则存在着一条导致接受态的计算路径. 若猜测模块给出了该计算路径，则验证模块的终结状态为接受态 q_Y，并且验证时间为多项式的.

对于 NP 类，相当于要求验证模块可在多项式时间内完成(对于导致接受的计算路径)，反之亦然.

例如，判断一个图是否有哈密顿(Hamilton)回路就属于 NP 类. 因为若图 G 有

哈密顿回路,且当给出了该回路,则显然可在多项式时间内予以验证.

以上定义了确定型和非确定型图灵机,以及 P 和 NP 类.这些都是 NP 理论中的基本概念,有了它们,才能够讨论算法的复杂度以及问题的难解程度.P 类问题是能够有效解决的,但 NP 类问题则不一定,例如判断哈密顿回路问题.目前最好的算法仍需要指数量级的时间.至于 NP 类中的问题是否都能有效地解决,即 NP 是否等于 P,这是 NP 理论中的核心问题,迄今尚未得到解决.

虽然有时不能证实一个问题的难度,但可以比较问题的难度.从这一点出发,也可以得到许多优美、重要的结果.

8.3　多项式归约

当比较 A、B 两个问题的难度时,常常会用到归约的思想.若问题 A 可有效地归约为对问题 B 的计算,则问题 A 的难度将不会超过问题 B.在 NP 理论中,使用最多的是多项式归约这一手段,而正是多项式归约搭起了问题之间的桥梁.

设

$$\Pi \xlongequal{\text{def}} \{f \mid f: \{0,1\}^* \to \{0,1\}^*, \text{且 } f \text{ 的计算可在多项式时间内完成}\}.$$

Π 中的元素称作是 $\{0,1\}^*$ 上的多项式变换.

定义2　设 $A \subseteq \{0,1\}^*$, $B \subseteq \{0,1\}^*$,若存在 $f \in \Pi$,使得 $x \in A \Leftrightarrow f(x) \in B$,则称问题 A 可多项式归约为问题 B,并表示为 $A \propto B$.

若 $A \propto B$,则 A 的难度将不超过 B 的难度,因为若 B 解决了,则 A 通过归约的方法也能够得到解决.事实上,有以下引理:

引理　若 $A \propto B$,且 $B \in \mathrm{P}$,则 $A \in \mathrm{P}$.

证明　设 f 是定义中的多项式变换,则存在多项式 g,使 $f(x)$ 的计算时间至多为 $g(|x|)$($|x|$ 表示串 x 的长度).由于 f 的计算时间不超过 $g(|x|)$,则其输入长度至多为 $g(|x|)$.

又由于 $B \in \mathrm{P}$,故存在一多项式时间的确定型图灵机 TB 识别 B,设该多项式界为 h.如图 8-3 构造图灵机 TA.

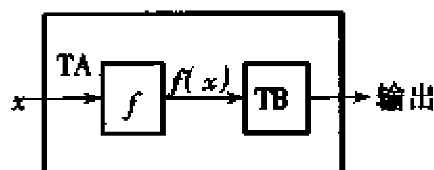


图 8-3

由于当且仅当 $f(x) \in B$ 才 $x \in A$,即当且仅当 $f(x)$ 为 TB 所接受,因而上述 TA 可识别 A,且 TA 的运行时间至多为 $g(|x|) + h(g(|x|))$,仍为 $|x|$ 的多项式,因而 $A \in \mathrm{P}$.

以上过程也可表达为图 8-4 所示.

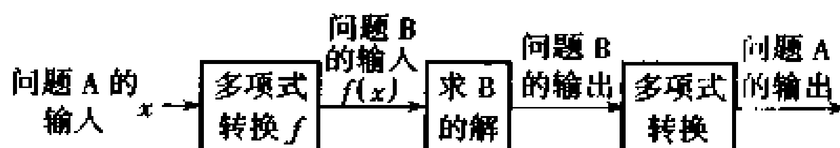


图 8-4

另外，多项式归约还具有自反性及传递性，即

1° $A \propto A$；

2° 若 $A \propto B$、$B \propto C$，则 $A \propto C$.

这些性质利用"$\propto$"的定义易证，由读者自己来完成.

有了多项式归约，就可以比较问题的难度，建立问题之间的联系. 在一类问题中，最令人感兴趣的是那些最复杂的问题，因为这些问题代表着这类问题难度的上界，一旦它们得到了有效解决，则整类问题都可迎刃而解. 这些最难的问题通常称为某类的完全问题，最著名的是 NP 完全问题，即 NP 类中最难的问题. NP 完全类(也称 NP 完备类，简称 NPC)定义如下：

$$\mathrm{NPC} \xlongequal{\mathrm{def}} \{C \mid C \in \mathrm{NP}, \forall A \in \mathrm{NP}, \text{使 } A \propto C\}.$$

由多项式归约的定义，可见 NPC 类中的问题是 NP 类中最难的，因为 NP 类中所有的问题都可以归约到它们中的某一个. 对 NPC 问题的研究是 NP 理论中的精髓部分.

8.4 可满足性问题及库克定理

上节定义了 NP 完全问题的概念，但 NP 完全问题是否存在？若存在，它们是什么样的问题？它们是否的确很困难？这些都并未能在定义中得到阐明. 1971 年，库克(Cook)提出了可满足性问题，并证明了它是一个 NP 完全问题，奠定了 NP 理论的基础.

1. 可满足性问题(Satisfiability Problem, SAT)

先介绍几个必要的概念，一个逻辑表达式由逻辑变量和"与"、"或"、"非"几种运算构成，如

$$(\bar{A} \vee B \vee C) \wedge (\bar{A} \vee C \vee D) \wedge (B \vee \bar{C}) \wedge (\bar{A} \vee C \vee \bar{D}),$$

其中每个逻辑变量只取"真(T)"和"假(F)"两个值，对应于逻辑变量的每一种指派，可确定逻辑表达式的取值. 如当 $A =$ "F"，$B = C = D =$ "T"时，上述逻辑表达式取值为"T(真)".

任何一个逻辑表达式 f 都可写为以下形式：

$$f = C_1 \wedge C_2 \wedge \cdots \wedge C_k,$$

其中每个 C_i 均为逻辑变量("或"与"非")的逻辑或，称逻辑表达式 f 的这种形式为合取范式，其中每个 C_i 称为 f 的子句. 可见 f 取值为真当且仅当每个子句 C_i 取值都为真，即每个子句中至少有一个子项取值为真. 可满足性问题就是任给一个合取范式 f，问是否存在某种赋值使 f 取值为真. 它等价于下面的描述.

设 $L = \{A, B, \cdots, \bar{A}, \bar{B}, \cdots\}$，$C_1, C_2, \cdots, C_k$ 是 L 的有限子集，称为子句. 每个 C_i

中不出现 L 中互补的一对(即若 $x\in C_i$,则 $\bar{x}\bar{\in} C_i, i=1,2,\cdots,k$),所谓可满足性问题,是指任给了 k 个子句 $C_1,\cdots,C_k$,确定是否存在一集合 $S\subseteq L$,满足:

1° S 中不包含互补的一对元素,即若 $x\in S$,则 $\bar{x}\bar{\in} S$.

2° $S\cap C_i\neq\varnothing, i=1,2,\cdots,k$.

若这样的 S 存在的话,则可对逻辑变量适当指派,使得 S 中的每个元素取值为真(由于 1°,这总可以做到),由 2°知每个 C_i 都取值为真,因而整个逻辑表达式的取值为真.

NP 类中的每个问题都可多项式归约到可满足问题 SAT.这就是著名的库克定理.下面先举一个图的着色问题归约到可满足问题的例子.

图的着色问题是这样的:已知有限图 G 及整数 m,确定是否可用 m 种颜色对图 G 的顶点进行着色,使得相邻接的顶点有不同颜色.

定理1 图的着色问题可多项式归约到可满足问题 SAT.

证明 已知图 G 及整数 m,令 $C_1,C_2,\cdots,C_m$ 表 m 种颜色,设

$$p_{ij}=\begin{cases}T, & \text{若顶点 } i \text{ 着以颜色 } C_j\\ F, & \text{其它}\end{cases}$$

$$i=1,2,\cdots,n\ (n=|V|);j=1,2,\cdots,m$$

这样,G 的每一种着色方案对应于给 $n\times m$ 个变量 $\{p_{ij}\}$ 的一种指派.但是

(1) 每个顶点至少有一种颜色,故对于任一顶点 i,对应有子句 $p_{i1}\vee\cdots\vee p_{im}$.

(2) 相互邻接的顶点着以不同颜色,故对图 G 的任意一对相互邻接顶点(r, s),r、s 不能在一种着色方案下着以同一颜色,即有子句 $\overline{p_{rj}}\vee\overline{p_{sj}}, j=1,2,\cdots,m$.

可见 G 可用 m 种颜色进行着色使相互邻接顶点不同色,当且仅当存在一种对变量 $\{p_{ij}\}$ 的赋值,使得由(1)、(2)构成的子句取值均为真.

另外,上述子句的构造显然具有多项式界.因而定理得证.

2. 库克定理

库克定理是 NP 理论的重要支柱之一,定理的证明要用到数理逻辑中的一些恒等式,它们是

(1) $A\rightarrow B=\bar{A}\vee B$,

$A\leftrightarrow B=(A\rightarrow B)\wedge(B\rightarrow A)=(\bar{A}\vee B)\wedge(A\vee\bar{B})$.

(2) $\bar{\bar{A}}=A, \overline{A\vee B}=\bar{A}\wedge\bar{B}, \overline{A\wedge B}=\bar{A}\vee\bar{B}$.

(3) $A\vee(B\wedge C)=(A\vee B)\wedge(A\vee C)$.

用以上恒等式可逐步将任何一个逻辑表达式化为合取范式.例如:

$$\begin{aligned}\bar{A}\vee\overline{(B\rightarrow C)}\vee D&=\bar{A}\vee\overline{(\bar{B}\vee C)}\vee D\\&=\bar{A}\vee(\bar{\bar{B}}\wedge\bar{C})\vee D=(\bar{A}\vee(B\wedge\bar{C}))\vee D\\&=((\bar{A}\vee B)\wedge(\bar{A}\vee\bar{C}))\vee D=D\vee((\bar{A}\vee B)\wedge(\bar{A}\vee\bar{C}))\\&=(D\vee(\bar{A}\vee B))\wedge(D\vee(\bar{A}\vee\bar{C}))\\&=(D\vee\bar{A}\vee B)\wedge(D\vee\bar{A}\vee\bar{C}).\end{aligned}$$

定理 2(库克定理) 若 $L\in$ NP,则 $L\propto$ SAT.

证明 只需证明任何 NDTM 在多项式时间内接受的任一集合,可以通过多项

式界时间变换为可满足性问题.

设已知一 NDTM,记为 M,要在 $T(n)$(多项式)界内识别 L,只要找一有多项式界的函数 f 把输入符号串转换为一组子句 $f(x)$,而且 M 接受 $x \Leftrightarrow f(x)$ 为可满足的

设 NDTM 的有限状态集合 Q 为

$$Q=\{q_1 q_2,\cdots,q_l\},$$

其中 q_1 为初始状态,q_2 为 q_Y,q_3 为 q_N.

带字母表为

$$A=\{a_1,a_2,\cdots,a_m\},$$

其中 a_1 为空格符.

设机器在不超过 N 步内接受 x,每走一步,读写头最多左移或右移一格.故读写头的范围不超过以初始位置为中心的左右各 N 格,即至多共涉及 $2N+1$ 格.每行 $2N+1$ 格上的符号串、机器当时的状态以及读写头的位置,这些信息完整地描述了机器的格局.

t 从 1 到 N,每个时刻机器的格局是 x 被接受的全部过程.

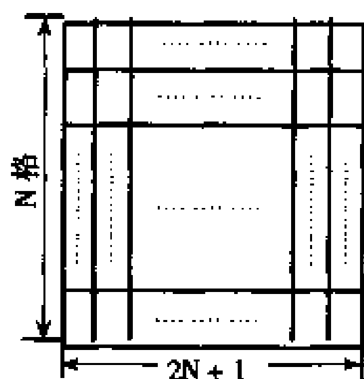


图 8-5

如图 8-5 所示,引进一 $N\times(2N+1)$的方格,图中第 t 行为时刻 t 上的符号,第 t 行第 i 列的方格用(t,i)表示.为了描述机器运行的全部过程,特引进以下变量:

$$A(t,i,j)=\begin{cases}T, & 若(t,i)的符号为 a_j,\\ F, & 其它,\end{cases}$$

$$H(t,i)=\begin{cases}T, & 若读写头 t 时刻位于第 i 格,\\ F, & 否则,\end{cases}$$

$$S(t,k)=\begin{cases}T, & 若在 t 时刻,机器状态为 q_k,\\ F, & 否则,\end{cases}$$

其中 $1\leqslant t\leqslant N,1\leqslant i\leqslant 2N+1,1\leqslant j\leqslant m,1\leqslant k\leqslant l$.

对应于任一输入 $x=a_{k_1}\cdots a_{k_n}$,定义以下几组子句:

(1) 对应于初始状态

1) 初始输入符号串为 $a_{k_1}\cdots a_{k_n}$,即图 8-5 的第一行为

$$\underbrace{\#\ \#\cdots\#}_{N}a_{k_1}\cdots a_{k_n}\underbrace{\#\ \#\cdots\#}_{N-n+1},$$

可用子句描述如下:

$$\bigwedge_{i=1}^{N}A(1,i,1)\bigwedge_{i=N+1}^{N+n}A(1,i,k_{i-N})\bigwedge_{i=N+n-1}^{2N+1}A(1,i,1).\tag{1}$$

2) 初始状态为 q_1,读写头位于 $N+1$ 格,对应子句为

$$S(1,1)\wedge H(1,N+1).\tag{2}$$

(2) 对应于任一时刻格局的唯一性(图像转移是不唯一的)

3) 对任一 $t,1\leqslant t\leqslant N$,机器状态是唯一的,对应子句为

$$\bigwedge_{t=1}^{N}\Big(\Big(\bigvee_{j=1}^{l}S(t,j)\Big)\wedge\Big(\bigwedge_{1\leqslant j_1<j_2\leqslant l}\overline{(S(t,j_1)\wedge S(t,j_2))}\Big)$$

$$= \bigwedge_{t=1}^{N}\left(\left(\bigvee_{j=1}^{l} S(t,j)\right) \bigwedge_{1\leqslant j_1<j_2\leqslant l} \overline{S(t,j_1)} \vee \overline{S(t,j_2)}\right). \tag{3}$$

4) 对任一 $t,i(1\leqslant t\leqslant N,1\leqslant i\leqslant 2N+1)$ 的任一方格 (t,i) 有一个字符，且仅有一个字符，和 3)类似，对应的子句为

$$\bigwedge_{t=1}^{N}\bigwedge_{i=1}^{2N+1}\left(\bigvee_{j=1}^{m} A(t,i,j) \bigwedge_{1\leqslant j_1<j_2\leqslant m} \overline{A(t,i,j_1)} \vee \overline{A(t,i,j_2)}\right). \tag{4}$$

5) 对任一 $t,1\leqslant t\leqslant N$，读写头在且仅在某一方格上，对应的子句为

$$\bigwedge_{t=1}^{N}\left(\bigvee_{i=1}^{2N+1} H(t,i) \bigwedge_{1\leqslant i_1\leqslant i_2\leqslant 2N+1} (\overline{H(t,i_1)} \vee \overline{H(t,i_2)})\right). \tag{5}$$

(3) 对应于图像转移

6) 设 t 时刻时机器状态为 q_k，读写头位于第 i 格，而第 i 格上对应的字符为 a_j，又设 $c=(q_k', a_j', \text{action})$，$q_k'\in Q$，$a_j'\in A$，action$\in\{h,r,l\}$. 令

$$k_{c_1}=k',\quad 1\leqslant k\leqslant l,$$

$$j_{c_2}=j',\quad 1\leqslant j\leqslant m,$$

$$i_{c_3}=\begin{cases} i, & \text{若 action}=h,\\ i+1, & \text{若 action}=r,\\ i-1, & \text{若 action}=l,\end{cases}\qquad 1\leqslant i\leqslant 2N+1,$$

则每次图像转移对应于以下子句：

$$(A(t,i,j)\wedge H(t,i)\wedge S(t,k))\longrightarrow$$

$$\bigvee_{c\in f(q_k,a_j)}(A(t+1,i,j_{c_2})\wedge H(t+1,i_{c_3})\wedge S(t+1,k_{c_1}))$$

$$=\overline{A(t,i,j)\wedge H(t,i)\wedge S(t,k)}\vee\left(\bigvee_{c\in f(q_k,a_j)}(A(t+1,i,j_{c2})\wedge H(t+1,i_{c_3})\wedge S(t+1,k_{c_1}))\right)$$

$$=\overline{A(t,i,j)}\vee\overline{H(t,i)}\vee\overline{S(t,k)}\vee\left(\bigvee_{c\in f(q_k,a_j)}(A(t+1,i,j_{c_2})\wedge H(t+1,i_{c_2})\wedge S(t+1,k_{c_1}))\right).$$

由于 $|f(q_k,a_j)|$ 为一常数，故上式可在常数步内化为合取范式. 对于机器的所有图像转移有

$$\bigwedge_{t=1}^{N}\bigwedge_{i=1}^{2N+1}\bigwedge_{k=1}^{l}\bigwedge_{j=1}^{m}(\overline{A(t,i,j)}\vee\overline{H(t,i)}\vee\overline{S(t,k)})$$

$$\vee\left(\bigvee_{c\in f(q_k,a_j)}(A(t+1,i,j_{c_2})\wedge H(t+1,i_{c_2})\wedge S(t+1,k_{c_1}))\right). \tag{6}$$

(4) 对应于终态

7) 对应于终结接受态有

$$S(N,2). \tag{7}$$

对以上一组子句((1)～(7))做"∧"运算得一合取范式,则该逻辑表达式可满足,当且仅当存在一种指派使各子句都取值为真,这等价于存在一条计算路径使得M在 N 步内接受 x. 由于 1、m 均为常数,而 N 为输入长度 x 的多项式,故整个合取范式的长度不会超过 x 长度(n)的某个多项式的取值,因而上述转换过程为一多项式变换,从而证明了对所有 $L\in$ NP 类,$L\propto$ SAT. 证毕.

另外,有引理

引理 SAT∈NP.

证明 显然,若一个逻辑表达式可满足,则若给出了使得其取值为真的那组赋值,则可在多项式时间内予以验证,故 SAT∈NP.

于是由上述两个结论知 SAT∈NPC,即 SAT 问题为一 NP 完全问题.

8.5 若干 NP 完全问题及其证明

继库克证明 SAT 问题是 NPC 的以后,1972 年,卡普又给出了若干 NP 完全问题,这些问题的 NP 完全性证明思想和技巧以及利用它们证明的大批 NPC 问题,极大地丰富了 NP 理论,可以说,卡普和库克的工作共同奠定了 NP 理论的基础.

要证明一个问题 P_1 是 NP 完全的,只需证明以下两点:

(1) $P_1\in$ NP.

(2) 存在一个 NPC 问题 P_2,有 $P_2\propto P_1$(由 $\propto$ 的传递性).

下面将介绍若干 NPC 问题,它们是 3SAT 问题、团问题、图着色问题、独立集问题、顶点覆盖问题.

8.5.1 3SAT 问题

对于合取范式

$$f=C_1\wedge C_2\wedge\cdots\wedge C_n,$$

若每一子句 C_i 所含的文字数目都恰为 3 时,它的可满足性问题便称为**三元可满足问题**(3SAT). 只要证明任一逻辑表达式都可化为等价的满足上述要求的逻辑表达式,即可证 3SAT∈NPC.

关于可满足性,有以下简单结果:

(1) λ 可满足当且仅当$(\lambda\vee y_1\vee y_2)\wedge(\lambda\vee\overline{y_1}\vee y_2)\wedge(\lambda\vee y_1\vee\overline{y_2})\wedge(\lambda\vee\overline{y_1}\vee\overline{y_2})$可满足.

(2) $\lambda_1\vee\lambda_2$ 可满足当且仅当$(\lambda_1\vee\lambda_2\vee y)\wedge(\lambda_1\vee\lambda_2\vee\overline{y})$可满足.

(3) $f_1\vee f_2$ 可满足当且仅当$(y\vee f_1)\wedge(\overline{y}\vee f_2)$可满足,其中 y 不在 f_1,f_2 中出现.

利用这些结果,可证以下定理:

定理3 3SAT∈NPC.

证明 (1) 由于 3SAT 是 SAT 的特殊情形,故有 3SAT∈NP.

(2) 下证 SAT$\propto$3SAT. 任给逻辑表达式 $f=C_1\wedge\cdots\wedge C_n$,对每个 C_i 作以下变换:

1) 若 C_i 恰为三个文字，则 C_i 不变.

2) 若 C_i 不为三个文字，则利用上述(1)、(2)、(3)引进适量的辅助逻辑变量将其化为每个子句恰含三个文字的合取范式，且它可满足当且仅当 C_i 可满足.

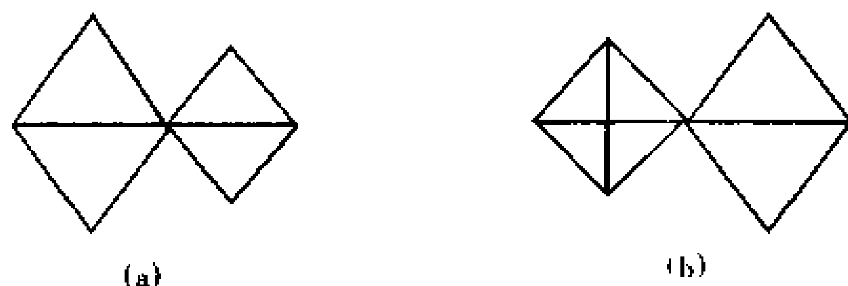
由 1)、2)得到的合取范式 f' 满足：① f' 中每个子句中文字的数目不超过三个. ② f 可满足当且仅当 f' 可满足，且由 f 到 f' 的变换过程可在多项式时间内完成. 故 SAT∝3SAT.

由(1)，(2)有 3SAT∈NPC，证毕.

8.5.2 团问题

若完全图 G_1 是图 G 的子图，则称图 G_1 是图 G 的团.

如图 8-6(a)中有若干个 3 个顶点的团，但没有超过 3 个顶点的团，图(b)中有 4 个顶点的团.



(a) (b)

图 8-6

团问题：任给图 G 和整数 k，试判定图 G 中是否存在 k 个顶点的团？

定理4 团问题∈NPC.

证明 同样分两部分进行.

(1) 证团问题∈NP. 显然，验证 G 的一个子图是团只需验证需多项式时间即可.

(2) 证 SAT∝团问题. 任给表达式 f，构造一个相应的图 G 如下：

1) 图 G 的每个顶点对应于 f 中的每个文字(多次出现的重复计算).

2) 若 G 中两个顶点其原对应的文字不相互补且不出现于同一子句中，则将其连线.

例如公式 $(A \vee \bar{B}) \wedge (B \vee \bar{C}) \wedge (C \vee \bar{A})$ 对应于图 8-7. 图 8-7 等价于图 8-8.

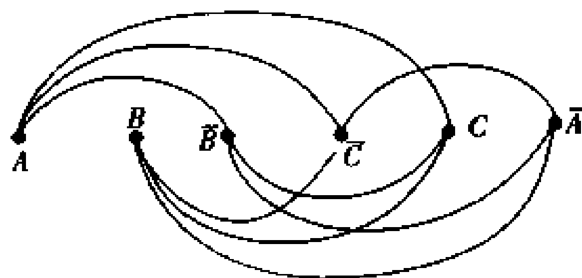


图 8-7

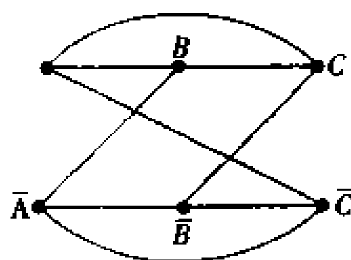


图 8-8

设 f 有 n 个子句，则 f 可满足当且仅当 f 对应的图 G 中有 n 个顶点的团. 这是因为①若 f 可满足，即有某种赋值使得 f 取值为真，它等价于使得每个 C_i 中都至少有一个文字为真，这 n 个文字(每个 $C_i, 1 \leqslant i \leqslant n$ 中一个)对应的图 G 中的 n 个顶点就构成一个团. ②若图 G 中有一 n 个顶点的团，则取给出使得这 n 个顶点对应的文字都为真的指派，则 f 的取值为真(这由图 G 的定义易证).

显见，上述构造图 G 的方法是多项式界的，因此 SAT$\propto$团问题.

由(1)、(2)，团问题$\in$NPC，证毕.

8.5.3　图的着色问题

图的着色问题是指任给简单图 G 及正整数 k，问是否存在对图 G 顶点的 k 着色，使得 G 中任意两个相邻接的顶点都着以不同颜色?

可证明 SAT$\propto$图着色问题. 从而证图着色问题$\in$NPC.

定理5　图着色问题$\in$NPC.

证明　(1) 任给一种着色方案，容易判定它是否满足使得相邻接顶点着色不同，因而图着色问题$\in$NP.

(2) 证明 SAT$\propto$图着色问题. 设 $L = \{x_1, \cdots, x_n, \bar{x}_1, \cdots, \bar{x}_n\}$，而 f 为 L 上的合取范式，且 $f = C_1 \wedge C_2 \wedge \cdots \wedge C_m$，对应于 f，构造下述的图 G:

令 G 的顶点为

$$C_1, C_2, \cdots, C_m, x_1, x_2, \cdots, x_n, \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n, y_1, y_2, \cdots, y_n, z$$

其中 $y_1, \cdots, y_n, z$ 为引进的辅助顶点.

图 G 的边为

$(x_i, \bar{x}_i), i = 1, \cdots, n$,

$(x_i, y_j), i \neq j, i, j = 1, \cdots, n$,

$(\bar{x}_i, y_j), i \neq j, i, j = 1, \cdots, n$,

$(y_i, y_j), i \neq j, i, j = 1, \cdots, n$,

$(z, y_j), j = 1, \cdots, n$,

$(C_i, z), i = 1, \cdots, m$,

(x_i, C_j)，若 $x_i \bar{\in} C_j, i = 1, \cdots, n, j = 1, \cdots, m$,

$(\bar{x}_i, C_j)$，若 $x_i \bar{\in} C_j, i = 1, \cdots, n, j = 1, \cdots, m$.

则有 f 可满足当且仅当图 G 可 $n+1$ 着色.

证明如下:

1) 若图 G 可 $n+1$ 着色，由于 $y_1, y_2, \cdots, y_n$ 构成一完全图，故它们两两的着色均不同，不妨设 y_j 着以颜色 $j, j = 1, 2, \cdots, n$. 显然若 $i \neq j$，则 x_i 与 y_j 不可同色，故 x_i 只可着以颜色 i 或 $n+1$. 同理，$\bar{x}_i$ 也是这样，但 x_i、$\bar{x}_i$ 不能着以相同颜色，故 x_i、$\bar{x}_i$ 中有一个着以颜色 i，一个着以颜色 $n+1, i = 1, 2, \cdots, n$. 令 $\lambda_i \in \{x_i, \bar{x}_i\}$，且 λ_i 着以颜色 $n+1, i = 1, 2, \cdots, n$. 取使得 $\lambda_i (i = 1, 2, \cdots, n)$ 均为假的指派，则在该指派下 f 取值为真. 这是因为，由于 z 与 $y_j (j = 1, \cdots, n)$ 相连，故 z 只能取第 $n+1$ 种颜色，因而任一 C_i 都不能取颜色 $n+1$. 不妨设 C_i 着以颜色 $j \leqslant n$，而 x_j、$\bar{x}_j$ 中至少有一个与

C_i 相连(因为 x_j、$\bar{x}_j$ 不能同在 C_i 中),设为 $\lambda'_j \in \{x_j, \bar{x}_j\}$,而 λ'_j 只能着以颜色 j 或 $n+1$,故 λ'_j 必着以颜色 $n+1$,且 $\overline{\lambda'_j}$ 着以颜色 j,这说明 $\overline{\lambda'_j}$ 不与 C_i 相连,即 $\overline{\lambda'_j} \in C_i$,且 λ'_j 的取值为假,因此 C_i 取值为真. $C_1, C_2, \cdots, C_m$ 取值均为真,于是 f 取值亦为真,即 f 可满足.

2) 若 f 可满足,设令其取值为真的指派为 $\lambda_i = F, i = 1,2,\cdots,n, \lambda_i \in \{x_i, \bar{x}_i\}$,这时将图 G 中的 $y_1, y_2, \cdots, y_n$ 分别赋以颜色 $1,2,\cdots,n$, z 赋以颜色 $n+1$, λ_1 赋以颜色 i, $\overline{\lambda_i}$ 赋以颜色 $n+1(i=1,2,\cdots,n)$. 对于 $C_i(i=1,2,\cdots,m)$,由于 C_i 均取值为真,故有 $\lambda'_j \in C_i(\lambda'_j$ 为 x_j 或 $\bar{x}_j)$ 满足 λ'_j 取值为真,而 C_i 不与 λ'_j 相连,由上面着色的定义知 λ'_j 拥有颜色 j,将 C_i 赋以颜色 j. 这样就给出了图 G 的一种 $n+1$ 着色的方案,从而图 G 可 $n+1$ 着色.

由 1)、2)知,SAT $\propto$ 图着色问题.

以上即证图着色问题 $\in$ NPC. 证毕.

8.5.4 独立集问题

设图 $G=(V,E)$, S 是顶点 V 的子集,若集合 S 中的任意两个顶点均不相邻,则称 S 为图 G 的**独立集**.

独立集问题:任给图 $G=(V,E)$ 及正整数 $k \leqslant |V|$,判断 G 中是否有 k 个顶点的独立集?

作 G 的补图 $\bar{G}=(V,\bar{E})$,即 $(u,v) \in \bar{E}$ 当且仅当 $(u,v) \notin E$. 则一个顶点集合 S 为 G 中某个团的顶点当且仅当 S 为 $\bar{G}$ 的独立集,故团问题 $\propto$ 独立集问题. 另外独立集问题 $\in$ NP,这样立即有

定理6 独立集问题 $\in$ NPC.

8.5.5 顶点覆盖问题

设有图 $G=(V,E)$,若 C 为 V 的子集,且 G 中任一条边都至少有一个顶点在 C 中,则称 C 为图 G 的一个**顶点覆盖**.

顶点覆盖问题:任给图 $G=(V,E)$ 及正整数 $k \leqslant |V|$,问 G 是否存在 k 个顶点的顶点覆盖?

对于一个图 $G=(V,E)$,若 $C \subseteq V$,则 C 为 G 的一个顶点覆盖当且仅当 $V \setminus C$ 为 G 的一个独立集. 即一个图 G 具有 k 个顶点的独立集当且仅当它具有 $|V|-k$ 个顶点的顶点覆盖. 因而独立集问题 $\propto$ 顶点覆盖问题,而顶点覆盖问题 $\in$ NP,故有

定理7 顶点覆盖问题 $\in$ NPC.

8.6 复杂度类

算法复杂性理论是将问题的难度进行分类. P 类是可在多项式时间内求解的一类, NP 类是在多项式时间内对问题进行验证的一类. 显然

$$P \subseteq NP.$$

人们感兴趣的是 P = NP? 若 P = NP,则所有的 NP 类,其中包括 NPC 类问题都有多项式算法.若 NP≠P,则肯定了 NP 类存在许多问题,其中包括 NPC 类问题不存在多项式算法.不论哪一种结论都将是对算法研究的极为重大突破.

遗憾的是这个 NP 理论的重要问题还未得到解决.

定理8　$P = NP$ 当且仅当 $P \wedge NPC \neq \varnothing$.

P 是 NP 类中最容易的一类,NPC 是其中最难的.

CO-NP 问题:

与 SAT 问题相反,任给一合取范式,证明它不可满足.这样的问题还不清楚是否属于 NP 类.

$L \subseteq \{0,1\}^*$, $\bar{L} = \{0,1\}^* \setminus L$,则关于 $\bar{L}$ 问题的判定问题称为 L 问题的余问题.NP 类中所有问题的余问题构成的集合称为 CO-NP 类,即

$$\text{CO-NP} = \{\bar{L} \mid L \in \text{NP}\}.$$

SAT 问题的余问题属于 CO-NP 类.$P \leqslant \text{CO-NP}$,但 NP 和 CO-NP 的关系还不清楚.

不难证明

$$L \leqslant \text{NPC} \quad \text{当且仅当 } L \in \text{CO-NPC}.$$

因此 SAT 问题的余问题是 CO-NP 完全问题.

NPH 问题:

若一问题,所有 NP 问题均可归约为它,则该问题称为 NPH 或 NP 难题.它与 NPC 问题的区别在于它不必属于 NP 类.

$$\text{NPC} \leqslant \text{NPH}.$$

流动推销员问题不是 NP 类,它的难度与 NPC 问题相当,属于 NPH 类.

参 考 文 献

1　Thomas H C, Charleo E L, Ronald L R. Introduction to algorithms. Cambridge: MIT Press, 1992
2　卢开澄.算法与复杂性.北京:高等教育出版社,1995.
3　卢开澄.计算机算法导论——设计与分析.北京:清华大学出版社,1996.

·计算机数学卷·

第13篇

组合最优化的近似算法

编　者　张立昂
审校者　耿素云

目　录

引　言

NP 完全性理论成功地描述了包括流动推销员、装箱、图着色等在内的一大批组合最优化问题的计算难度．除非 P = NP,否则所有 NP 难的问题都不存在多项式时间算法．但是,这并不意味着事情的终结．由于在实际中的广泛应用,寻找求解 NP 难问题的有效算法的努力不会因此而终止．相反,NP 完全性理论刺激人们以更大的热情对 NP 难问题进行深入的研究．近似算法就是解决 NP 难问题的一条有效途径．

事实上,在 NP 完全性理论诞生之初,人们就开始把目光注视到近似算法上．约翰逊(D.S.Johnson)在 1974 年发表的一篇论文中系统地研究了一批近似算法的性能．接着桑尼(S.Sahni)与冈泽利兹(T.Gonzalez)于 1976 年证明流动推销员问题等不存在具有常数比的多项式时间近似算法(除非 P = NP),从而奠定了近似算法研究的两个主要方向:(1)设计有效的近似算法,并分析它的近似性能;(2)问题的不可近似性．二十多年来,近似算法一直成为人们研究的热点,取得了丰硕的成果．

1　贪婪算法

1.1　贪婪算法

贪婪算法是一种最简单、最直接的算法设计技术．用贪婪算法(greedy algorithm)可以求得很多组合最优化问题的最优解．如,求最小生成树的库鲁斯卡(J.B.Kruskal)算法、在带非负权的图中求从给定顶点到其它各顶点最短路的戴克斯徒拉(E.W.Dijkstra)算法都是贪婪算法．

例 1　付钱问题．设有10元、5元、2元、1元4种货币．每种货币的数量充足．要付给顾客 a 元,a 是正整数．问如何付钱才能使付给顾客的货币张数最少?

直观上,面额越大,需要的张数越少．采用贪婪算法,一张一张地付给顾客,直到给足 a 元为止,每次都给面额不超过余额的最大货币．记 $c_1 = 10$, $c_2 = 5$, $c_3 = 2$, $c_4 = 1$. 令

$$a_1 = a,\quad x_1 = \lceil a_1/c_1 \rceil,$$
$$a_i = a_{i-1} - c_{i-1}x_{i-1},\quad x_i = \lceil a_i/c_i \rceil,\quad i = 2,3,4.$$

则 x_1、x_2、x_3、x_4 分别是付给顾客的 10 元、5 元、2 元、1 元的张数．

可以证明上述方法确实给出问题的最优解．但是,如果在上述付钱问题中添加一种4元货币,上述贪婪算法就不能保证得到最优解．例如,当 $a = 8$时,用贪婪

算法得到的解是:1张5元、1张2元和1张1元,共3张．而最优解是2张4元．因此,对于用贪婪思想设计的组合最优化问题的算法必须检查它是否能够保证给出最优解,证明它是或者用反例说明它不是问题的最优化算法．这种最优性的证明往往不是很简单的．

下面几个例子中的贪婪算法都是问题的最优化算法．

例2 分数背包问题．设有 n 件物品和一个背包,物品 i 的体积为 $a_i > 0$,价值为 $c_i > 0$,背包的容量为 $b > 0$. 物品是可分割的,物品 i 的 x 部分($0 \leqslant x \leqslant 1$)的体积为 $a_i x$,价值为 $c_i x$. 问:如何装才能使装入背包的物品总价值最大?即在约束条件

$$\sum_{i=1}^{n} a_i x_i \leqslant b$$

下,使目标函数

$$\sum_{i=1}^{n} c_i x_i$$

最大,这里 $0 \leqslant x_i \leqslant 1, i = 1,2,\cdots,n$.

常识告诉我们,应该尽可能地先装单位体积价值大的物品．

算法 FRACTION KNAPSACK

步1 按照 c_i/a_i 从大到小排列物品,不妨设 $c_1/a_1 \geqslant c_2/a_2 \geqslant \cdots \geqslant c_n/a_n$.

步2 for $i = 1,2,\cdots,n$ do
```
      if b > 0 then
          if a_i < b then x_i ← 1, b ← b − a_i
                  else x_i ← b/a_i, b ← 0
          else x_i ← 0.
```

例3 最小完成时间和的作业时间表．设有 n 个待处理的作业,每次只能处理一个作业,作业 i 的处理时间为 $t_i > 0$,完成作业 i 时的时间称作作业 i 的完成时间．如何安排作业的处理顺序使得所有作业的完成时间之和最小?

例如,有3个作业,$t_1 = 4, t_2 = 10, t_3 = 3$,共有6个处理方案,不妨设开始的时间为0,各方案的完成时间之和如下:

方案	顺序	完成时间之和
①	1,2,3	$4+(4+10)+(4+10+3)=35$
②	1,3,2	$4+(4+3)+(4+3+10)=28$
③	2,1,3	$10+(10+4)+(10+4+3)=41$
④	2,3,1	$10+(10+3)+(10+3+4)=40$
⑤	3,1,2	$3+(3+4)+(3+4+10)=27$
⑥	3,2,1	$3+(3+10)+(3+10+4)=33$

方案⑤是最优的．

假设前面已处理的作业依次是 $j_1, j_2, \cdots, j_{i-1}$,第 i 次处理作业 k,它的完成时间是

$$T_i = t_{j_1} + t_{j_2} + \cdots + t_{j_{i-1}} + t_k.$$

为了使完成时间之和最小,在这一步要使和的增加最小,即 T_i 最小. 由于前面的处理顺序已经确定,这只需使 t_k 最小. 因此,在每一次处理尚未处理的作业中选处理时间最小的作业,即按照 t_i 从小到大的顺序处理. 对于前面的例子,算法给出方案 ⑤.

例 4 带限期的作业时间表. 有 n 个待处理的作业,每个作业的处理时间均为 1,每次只能处理 1 个作业. 完成作业 i 的最后限期是 $d_i > 0$. 如果按期完成作业 i,则可获得利润 $p_i > 0$,总利润等于所有按期完成的作业的利润之和. 如何安排作业的处理顺序,使获得的总利润最大?

例如,有 4 个作业,限期和利润如下:

i	1	2	3	4
d_i	2	1	1	2
p_i	50	10	30	15

可以有下述几种方案:

方案	顺序	利润
①	1,4	65
②	2,1	60
③	2,4	25
④	3,1	80
⑤	3,4	45
⑥	4,1	65

其中方案 ④ 是最优的.

记 $S = \{1,2,\cdots,n\}$ 是 n 个作业的集合,$J \subseteq S$. 如果存在一个安排使 J 中的作业都能按时完成,则称 J 是一个可行解,它的利润 $p(J) = \sum_{j \in J} p_j$. 贪婪算法一步一步地进行,开始时取 $J = \varnothing$. 设第 i 步开始时已经得到可行解 J,为了在这一步得到利润尽可能大的可行解,把满足下述条件的不属于 J 的作业 j 加入 J:① $J \cup \{j\}$ 是可行解;② p_j 尽可能的大. 即取 j 使得

$$p_j = \max\{p_i \mid i \in S - J \text{ 且 } J \cup \{i\} \text{ 是可行解}\}.$$

若这样的 j 存在,则令 $J \leftarrow J \cup \{j\}$,进入下一步;若这样的 j 不存在,则计算结束.

剩下的问题是,如何判断一个作业子集合是否是可行解. 下述引理解决了这个问题.

引理 1 设 J 是 k 个作业的集合,$\pi = (i_1, i_2, \cdots, i_k)$ 是 J 中作业的一个排列且满足 $d_{i_1} \leqslant d_{i_2} \leqslant \cdots \leqslant d_{i_k}$,则 J 是可行解的充分必要条件是按 π 的顺序进行处理,J 中每一个作业都能在限期内完成,即 $t \leqslant d_{i_t}, t = 1,2,\cdots,k$.

算法 JS1:

输入:n 个作业的限期 $d_i \in \mathbf{Z}^+$ 和利润 $p_i \in \mathbf{Z}^+, i = 1,2,\cdots,n$.

输出：最优解 $J = \{j_1, j_2, \cdots, j_k\}$，且 $d_{j_1} \leqslant d_{j_2} \leqslant \cdots \leqslant d_{j_k}$.

步1　按 p_i 从大到小重新排列作业．不妨设 $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$.

步2　令 $J \leftarrow \varnothing, k \leftarrow 0, d_0 \leftarrow 0, j_0 \leftarrow 0$.

步3　for $m = 1, 2, \cdots, n$ do

(3-1) 找到 $i = \max\{t \mid 0 \leqslant t \leqslant k$ 且 $d_{j_t} \leqslant d_m\}$.

(3-2) if $d_m \geqslant i + 1$ 且 $d_{j_t} \geqslant t + 1 (i + 1 \leqslant t \leqslant k)$ then 把 m 插入 j_i 和 j_{i+1} 之间，且令 $k \leftarrow k + 1$.

算法 JS1 的运行时间为 $O(n^2)$.

下述引理2提供另一个判断作业子集合可行性的更有效方法．

引理2　作业集合 J 是可行的，当且仅当如下安排 J 中的作业可以使 J 中的作业都在限期内完成：依次把每一个作业 $j \in J$ 安排在时刻 t 执行，其中

$$t = \max\{t' \mid 0 < t' \leqslant \min\{n, d_j\}，且时刻\ t'\ 尚未被占用\}.$$

换句话说，依次考虑每一个作业 $j \in J$，尽可能晚地执行作业 j，只要不超过它的限期．如果有一个作业在它的限期前已没有自由时间（即尚未被占用的时间），则 J 不是可行解；否则 J 是可行解．当 J 是可行解时，这样安排可能会出现空隙．如果需要的话，不难压缩这个安排，消除这些空隙．

根据引理2，可以设计一个算法试图一个一个地填满长度 $l = \min\{n, d_{\max}\}$ 的序列，其中 $d_{\max} = \max\{d_i \mid 1 \leqslant i \leqslant n\}$．对于每一个时刻 t，令

$$n_t = \max\{i \leqslant t \mid 时刻\ i\ 是自由的\}.$$

把 $L = \{1, 2, \cdots, l\}$ 划分成若干个集合，使得 i 和 j 属于同一个集合当且仅当 $n_i = n_j$．对给定的集合 $A \subseteq L$，记 $F(A)$ 为 A 中的最小元素．此外，引入虚构的时刻0，且时刻0永远是自由的．

算法 JS2：

输入：n 个作业的限期 $d_i \in \mathbf{Z}^+$ 和利润 $p_i \in \mathbf{Z}^+, i = 1, 2, \cdots, n$.

输出：最优解 $J = \{j_1, j_2, \cdots, j_k\}$，且 $d_{j_1} \leqslant d_{j_2} \leqslant \cdots \leqslant d_{j_k}$.

步1　按 p_i 从大到小重新排列作业，不妨设 $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$.

步2　令 $l \leftarrow \min\{n, d_{\max}\}$，其中 $d_{\max} = \{d_i \mid 1 \leqslant i \leqslant n\}$.

步3　令 $j_i \leftarrow 0, F(\{i\}) \leftarrow i, i = 0, 1, \cdots, l$．把 $\{0, 1, \cdots, l\}$ 划分成 $l + 1$ 个集合，每一个集合含一个元素．

步4　for $i = 1, 2, \cdots, n$ do

(4-1) 找到包含 $\min\{n, d_i\}$ 的集合，设为 A.

(4-2) if $F(A) > 0$ then

(4-2-1) 令 $j_{F(A)} \leftarrow i$.

(4-2-2) 找到包含 $F(A) - 1$ 的集合，设为 B.

(4-2-3) 合并 A 和 B 为 C，令 $F(C) \leftarrow F(B)$.

步5　删去 $\{j_1, j_2, \cdots, j_l\}$ 中的0，把它压缩成 J.

算法 JS2 的运行时间为 $O(n \operatorname{lb} n)$.

1.2 拟 阵

设 E 是(非空)有限集合,$\mathscr{I}$ 是 E 的子集族,并且 $\mathscr{I}$ 在包含关系下是封闭的,即 $A \in \mathscr{I}$ 且 $A' \subseteq A$,蕴涵 $A' \in \mathscr{I}$,称 $S = (E,\mathscr{I})$ 是一个**子集系统**,$\mathscr{I}$ 的成员称作**独立集**.

设 $A \subseteq E, I \in \mathscr{I}$. 若 $I \subseteq A$,则称 I 是 A 的独立集. 若 I 是 A 的独立集且不存在 A 的独立集真包含 I,则称 I 是 A 的**极大独立集**.

关于子集系统 $S = (E,\mathscr{I})$ 的**组合最优化问题**是:对每一个 $e \in E$ 给定权 $w(e) \geqslant 0$,求独立集 $I \in \mathscr{I}$ 使 $\sum_{e \in I} w(e)$ 最大.

关于子集系统 $S = (E,\mathscr{I})$ 的组合最优化问题的贪婪算法的一般形式如下:

算法 GREEDY:

输入:有限集合 E 以及权 $w(e) \geqslant 0 (\forall e \in E)$.

输出:独立集 $I \subseteq E$.

步 1 令 $I \leftarrow \varnothing$.

步 2 while $E \neq \varnothing$ do

(2-1) 找到 E 中权最大的元素 e.

(2-2) 令 $E \leftarrow E - \{e\}$.

(2-3) if $I \cup \{e\} \in \mathscr{I}$ then $I \leftarrow I \cup \{e\}$.

定义 1 设 $M = (E,\mathscr{I})$ 是一个子集系统,如果对于 M 的组合最优化问题的每一个实例用贪婪算法都能求得最优解,则称 M 是一个**拟阵**(matriod).

定理 1 设 $M = (E,\mathscr{I})$ 是一个子集系统,下述命题是等价的:

(1) M 是一个拟阵.

(2) 若 I、$J \in \mathscr{I}$ 且 $|J| = |I| + 1$,则存在 $e \in J - I$,使得 $I \cup \{e\} \in \mathscr{I}$.

(3) 若 $A \subseteq E$,I 和 I' 是 A 的极大独立集,则 $|I| = |I'|$.

定义 2 设 $M = (E,\mathscr{I})$ 是一个拟阵,$A \subseteq E$. A 的极大独立集中的元素个数称作 A 在 M 中的**秩**,记作 $r(A)$.

E 的极大独立集称作**基**.

由 $\mathscr{I}$ 关于包含关系的封闭性,拟阵 $M = (E,\mathscr{I})$ 由它的基集合

$$\mathscr{B} = \{B \subseteq E \mid B \text{ 是 } M \text{ 的基}\}$$

完全确定,即

$$\mathscr{I} = \{I \mid \text{存在 } B \in \mathscr{B} \text{ 使 } I \subseteq B\}.$$

定理 2 设 E 是(非空)有限集合,$\mathscr{B}$ 是 E 的子集族,则 $\mathscr{B}$ 是 E 上拟阵的基集合,当且仅当 $\mathscr{B}$ 满足下述两条:

1° 若 B_1、$B_2 \in \mathscr{B}$,则 $|B_1| = |B_2|$.

2° 若 B_1、$B_2 \in \mathscr{B}$,则对每一个 $e \in B_1$,存在 $f \in B_2$ 使 $(B_1 - \{e\}) \cup \{f\} \in \mathscr{B}$.

对于例 4 中带限期的作业时间表问题,设 S 是作业集合,

$$\mathscr{I} = \{J \subseteq S \mid J \text{ 是一个可行解}\},$$

则 $M=(S,\mathscr{I})$ 构成一个拟阵．带限期的作业时间表问题是它的组合最优化问题，每个作业 j 的权是它的利润 p_j．

例 5　图的圈拟阵．设无向图 $G=(V,E)$．令

$$\mathscr{F}=\{F\subseteq E\mid F \text{ 是 } G \text{ 的一个森林}\},$$

则 $M_G=(E,\mathscr{F})$ 构成一个拟阵，称作图 G 的**圈拟阵**．

对于任何边子集 $E_1\subseteq E$，E_1 的极大独立集是子图 $G_1=(V,E_1)$ 的生成森林，故 $r(E_1)=|V|-c(G_1)$，其中 $c(G_1)$ 是 G_1 的连通度．G 的生成森林是 M_G 的基．

M_G 的组合最优化问题是**最大权森林问题**：给定无向图 $G=(V,E)$ 以及每一条边 $e\in E$ 的权 $w(e)\geqslant 0$，求一个森林 $F\subseteq E$ 使 $\sum_{e\in F}w(e)$ 最大．

实际上，当 G 连通时，最大权森林问题等价于最小生成树问题．对每一条边 $e\in E$，令 $w'(e)=w_0-w(e)$，其中 $w_0>\max\{w(e)\mid e\in E\}$．对 G 的每一棵生成树 T，

$$\sum_{e\in T}w'(e)=(|V|-1)w_0-\sum_{e\in T}w(e).$$

从而关于权 w'，T 是最小生成树当且仅当关于权 w，T 是最大权树（森林）．

1.3　贪婪的启发式算法

对于很多组合最优化问题，贪婪算法不能保证得到最优解．在这种情况下得到一个贪婪的启发式算法．很多启发式算法是根据贪婪思想设计出来的，或者由这类算法改进得到的．

例 6　流动推销员问题的最邻近法．**流动推销员问题**（travelling salesman problem，简记作 TSP）：给定 n 个城市 $c_1,c_2,\cdots,c_n$，每一对 c_i 和 c_j 之间的距离 $d(i,j)>0$，且 $d(i,j)=d(j,i)$．经过每一个城市且每一个城市只经过一次的路线称作环游．求长度最短的环游．

最邻近法（NN）的做法如下：从某一个城市出发，在每一步取离当前所在城市最近的尚未到过的城市作为下一个城市，直到走遍所有的城市为止，最后回到出发的城市．

表面上看，最邻近法是很合理的，即使不能保证得到最短的环游，也应该能得到比较短的环游，但事实并非如此．

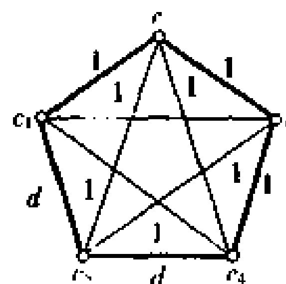


图 1-1

例如，图 1-1 有 5 个城市，其中 $d>1$．最优环游是 (c_1,c_3,c_5,c_2,c_4)，长度为 5．粗黑线是最邻近法给出的环游 (c_1,c_2,c_3,c_4,c_5)，长度是 $3+2d$．两条环游的长度之比为 $(3+2d)/5$．d 是任意一个大于 1 的数，这个比值可以任意的大．可见，最邻近法不仅只是一个近似算法，而且性能很差．

例 7　装箱问题的 FF 算法．**装箱问题**（bin packing problem）：给定 n 件物品 $u_1,u_2,\cdots,u_n$，物品 u_i 的体积为 $s_i\in(0,1]$．每一只箱子的容积为 1．要把 n 件物品全部装入箱子中，

求使用箱子最少的装法．

FF 算法(First Fit)十分直截了当．设想开始时顺序排列好空箱子 $B_1, B_2, \cdots$．按下述简单的规则一件一件地把所有物品装入箱子：在第 i 步把 u_i 装进剩余空间不小于 s_i 的下标最小的箱子．换句话说，总是把每一件物品放进第一只能装得下的箱子内．只有当所有已装有物品的箱子都装不下时才用一只新的空箱子．

例如装箱问题的实例 I 是有 $18m$ 件物品，它们的体积为

$$s_i = \begin{cases} \dfrac{1}{7} + \epsilon, & 1 \leqslant i \leqslant 6m, \\ \dfrac{1}{3} + \epsilon, & 6m < i \leqslant 12m, \\ \dfrac{1}{2} + \epsilon, & 12m < i \leqslant 18m, \end{cases}$$

其中 m 是一个正整数，$\epsilon > 0$ 且足够小．图 1-2(a)给出最优解，使用 $6m$ 只箱子，(b)是 FF 算法给出的近似解，使用 $10m$ 只箱子．

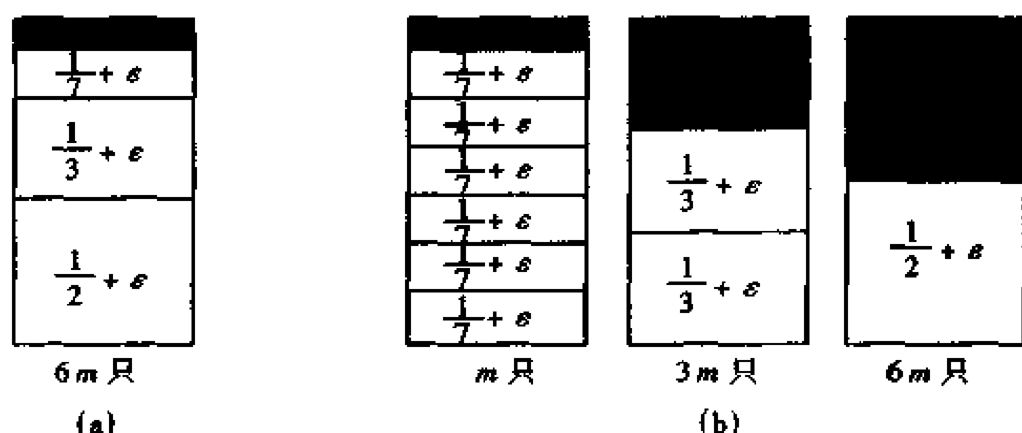


图 1-2

2　组合最优化的近似算法

2.1　近似算法的性能

很多组合最优化问题(combinatorial optimization problem)是 NP 难的，它们不存在多项式时间的最优化算法，除非 P = NP．例如，流动推销员问题、装箱问题、背包问题、图的着色问题等都是 NP 难的．既然在多项式时间内不能保证得到最优解，只好求其次，找一个满意的可行解或近似解．其实，对于一个实际问题也不一定非要“最优的”不可，往往是“足够好的”就可以了．因此，近似算法是解决 NP 难的组合

最优化问题的一条重要途径．

组合最优化问题 Π 由3部分组成：

(1) 实例集 D.

(2) 对于每一个实例 $I\in D$，有一个有穷的非空集合 $S(I)$，$S(I)$ 的元素称作实例 I 的可行解．

(3) 对于每一个实例 $I\in D$ 和它的可行解 $\sigma\in S(I)$，有一个正整数 $m(I,\sigma)$，称作 σ 的值．

当问题 Π 是最小化问题(最大化问题)时，如果 $\sigma^*\in S(I)$ 使得对所有的 $\sigma\in S(I)$ 有

$$m(I,\sigma^*)\leqslant m(I,\sigma)\qquad (m(I,\sigma^*)\geqslant m(I,\sigma)),$$

则称 σ^* 是实例 I 的最优解，$m(I,\sigma^*)$ 称作实例 I 的最优值，记作 $\mathrm{OPT}(I)$．实例 I 的规模记作 $|I|$.

例如，流动推销员问题是最小化问题．它的实例包括 n 个城市 $c_1,c_2,\cdots,c_n$ 以及每一对城市之间的距离 $d(i,j)$；每一条环游是一个可行解，可行解的值等于环游的长度；长度最短的环游是最优解．

如果对于问题 Π 的每一个实例 I，算法 A 输出 I 的一个可行解 σ，则称 A 是 Π 的**近似算法**(approximation algorithm)，并且记 $A(I)=m(I,\sigma)$．如果对于问题 Π 的每一个实例 I，A 总给出实例 I 的最优解，则称 A 是问题 Π 的**最优化算法**(optimization algorithm).

设 A 是问题 Π 的近似算法．对于问题 Π 的每一个实例 I，当问题 Π 是最小化问题时，记

$$R_A(I)=\frac{A(I)}{\mathrm{OPT}(I)};$$

当问题 Π 是最大化问题时，记

$$R_A(I)=\frac{\mathrm{OPT}(I)}{A(I)}.$$

$R_A(I)$ 称作算法 A 关于实例 I 的**性能比**(performance ratio).

定义1 设 $r:\mathbf{Z}^+\to(1,+\infty)$，$A$ 是问题 Π 的多项式时间近似算法．如果对于问题 Π 的每一个实例 I，

$$R_A(I)\leqslant r(|I|),$$

则称算法 A 是 $r(n)$ 多项式时间近似算法．

如果问题 Π 存在 $r(n)$ 多项式时间近似算法，则称问题 Π 是**在 $r(n)$ 范围内可近似的**，或**可近似到 $r(n)$ 范围内**．

定义2 算法 A 以问题 Π 的实例 I 和 $\epsilon>0$ 作为输入．当 ϵ 固定时，把这个算法记作 A_ϵ．如果对于每一个 $\epsilon>0$，A_ϵ 是问题 Π 的 $(1+\epsilon)$ 多项式时间近似算法，则称 A 是一个**多项式时间近似方案**(polynomial-time approximation scheme，简记作 PTAS).

如果 A 是一个 PTAS 且存在多项式 p 使得对于每一个实例 I 和 $\epsilon>0$，A 的运行时间不超过 $P(|I|,1/\epsilon)$，则称 A 是一个**完全多项式时间近似方案**(fully polynomia l-

time approximation scheme,简记作 FPTAS).

PTAS 和 FPTAS 的区别在于,当 A 是 PTAS 时,对每一个固定的 $\varepsilon>0$,虽然 A_ε 是多项式时间的,但是时间界限可能与 $1/\varepsilon$ 呈指数关系,如 $n^{1/\varepsilon}$. 随着 ε 接近 0,时间界限增长得很快;而 FPTAS 则要求算法的时间界限与 $1/\varepsilon$ 的关系是多项式的.

2.2 流动推销员问题的近似算法

为方便起见,流动推销员问题表述成下述形式:给定 n 个顶点的无向完全图 G,其顶点集 $V=\{1,2,\cdots,n\}$,距离矩阵 $\boldsymbol{D}=[d(i,j)]$,其中 $d(i,j)=d(j,i)\geqslant 0$,且 $d(i,i)=0,1\leqslant i,j\leqslant n$. 求 G 的最短哈密顿圈(Hamiltonian cycle).

2.2.1 最邻近法(NN)

在上一章 1.3 节例 6 中,描述了流动推销员问题的最邻近法. 在那里已经看到最邻近法的性能比可以任意的大. 事实上,这是流动推销员问题本身所固有的性质. 也就是说,流动推销员问题不可能有性能比较好的(如性能比不超过某个常数的)多项式时间近似算法,除非 P = NP.

下面对流动推销员问题的实例附加一个条件,假设距离满足三角不等式,即对于所有的 $1\leqslant i,j,k\leqslant n$, $d(i,k)+d(k,j)\geqslant d(i,j)$. 称这样的问题是满足三角不等式的流动推销员问题. 实际上,这样的假设是合理的. 例如,用两点之间的最短路的长度作为它们之间的距离,则满足三角不等式.

下述定理表明对于满足三角不等式的流动推销员问题,最邻近法也不是一个好的近似算法.

定理 1 对于满足三角不等式的流动推销员问题的任何 n 个城市的实例 I,有

$$\mathrm{NN}(I)\leqslant\left(\frac{1}{2}\lceil \mathrm{lb}\,n\rceil+\frac{1}{2}\right)\mathrm{OPT}(I),$$

并且对任意的 $m>3$,存在 $n=2^m-1$ 个城市的实例 I,使得

$$\mathrm{NN}(I)>\left(\frac{1}{3}\mathrm{lb}(n+1)+\frac{4}{9}\right)\mathrm{OPT}(I).$$

2.2.2 插入法(INS)

算法的基本思想是构造一个圈序列,每次增加一个顶点,直到圈经过所有 n 个顶点为止. 把图 G 中经过顶点子集 V' 中所有顶点的圈称作 V' 上的子环游. 约定:由一个顶点组成的集合上的子环游就是这个顶点本身,由两个顶点组成的集合上的子环游是连接这两个顶点的边(沿这条边走两遍).

设 C 是一条子环游,$k\in V\setminus C$,这里 $V\setminus C$ 表示不在 C 上的所有顶点. 按下述方式把 k 插入 C,得到一个新的添加了一个顶点的子环游,记作 $\mathrm{Tour}(C,k)$:

(1) 若 C 上有两个或两个以上顶点,则删去 C 上的边 $\{i,j\}$,添加边 $\{i,k\}$ 和 $\{k,j\}$,其中边 $\{i,j\}$ 使

$$d(i,k)+d(k,j)-d(i,j)$$

最小;

(2) 若 C 上只有一个顶点 i,则 Tour(C,k)等于边$\{i,k\}$.

把 Tour(C,k)与 C 的长度之差称作在 C 中插入 k 的代价,记作 Cost(C,k). 当 C 中的顶点数大于等于 2 时,

$$\mathrm{Cost}(C,k)=\min\{d(u,k)+d(k,v)-d(u,v)\mid[u,v]\in C\}.$$

算法 INS:

输入:n 个顶点的无向完全图上的对称距离矩阵$[d(i,j)]$.

输出:环游 C_n.

步 1　任取一个顶点 $u_0\in V$,令 $C_1\leftarrow\{u_0\}$.

步 2　for $i=1,2,\cdots,n-1$ do

　　选择 $u_i\in V\setminus C_i$,令 $C_{i+1}\leftarrow\mathrm{Tour}(C_i,u_i)$.

在步骤 2 中,采用不同的标准选择插入的顶点 u_i,得到不同的插入算法.

给定子环游 C 和顶点 $u\in V$,定义 C 和 u 的距离

$$d(C,u)=\min\{d(v,u)\mid v\in C\}.$$

最近插入法(Nearst-INS)　选择插入顶点 u_i 使得

$$d(C_i,u_i)=\min\{d(C_i,p)\mid p\in V\setminus C_i\}.$$

最佳插入法(Best-INS)　选择插入顶点 u_i 使得

$$\mathrm{Cost}(C_i,u_i)=\min\{\mathrm{Cost}(C_i,p)\mid p\in V\setminus C_i\}.$$

最近插入法和最佳插入法的时间复杂度分别为 $O(n^2)$和 $O(n^2\mathrm{lb}n)$.

定理 2　对于满足三角不等式的流动推销员问题的任何 n 个城市的实例 I,有

$$\text{Nearst-INS}(I)\leqslant 2\left(1-\frac{1}{n}\right)\mathrm{OPT}(I),$$

$$\text{Best-INS}(I)\leqslant 2\left(1-\frac{1}{n}\right)\mathrm{OPT}(I),$$

并且当 $n\geqslant 6$ 时,存在使等号成立的实例 I.

2.2.3　MST 算法

算法　MST(罗森克兰兹(D.J.Rosenkrantz)等,1977):

输入:n 个顶点的无向完全图 G 上的对称距离矩阵$[d(i,j)]$.

输出:G 的一条哈密顿圈 C.

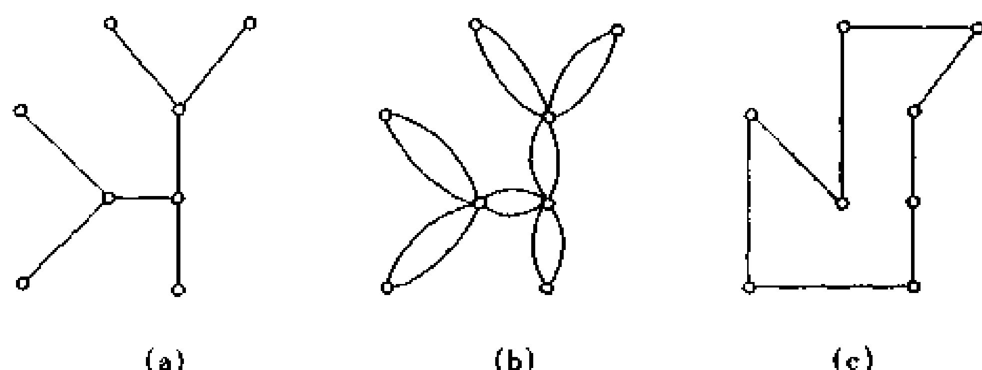
步 1　求图 G 的一棵最小生成树 T.

步 2　把 T 的每一条边复制成一对平行边,得到一个欧拉图,求这个欧拉图的欧拉迹.

步 3　沿欧拉迹跳过已经经过的顶点到下一个尚未经过的顶点,得到 G 的哈密顿圈 C.

MST 算法的计算过程如图 2-1 所示. 图(a)是求得的最小生成树 T,图(b)是由 T 得到的欧拉图及欧拉迹,图(c)沿欧拉迹抄近路得到 G 的哈密顿圈 C,即算法给出的近似解.

由于求欧拉迹只需要 $O(m)$步,这里 $m=2(n-1)$是步骤 2 中欧拉图的边数,



(a) (b) (c)

图 2-1

故算法的计算量基本上与求最小生成树的计算量相当．算法的时间复杂度为$O(n^2)$.

定理 3 对于满足三角不等式的流动推销员问题的所有实例 I,有

$$\mathrm{MST}(I) < 2\mathrm{OPT}(I),$$

并且对于任意的 $\epsilon > 0$,存在无穷多个实例 I 使得

$$\mathrm{MST}(I) \geq (2-\epsilon)\mathrm{OPT}(I).$$

2.2.4 MM 算法

在 MST 算法中,为了从最小生成树 T 得到一个欧拉图,把每一条边复制成一对平行边．实际上,只需将 T 中的奇度顶点两两配对(奇度顶点必有偶数个),在每一对之间加一条边,就可得到一个欧拉图．

算法 MM(克里斯托费兹(N. Christofides),1976):

输入:n 个顶点的无向完全图 G 上的对称距离矩阵$[d(i,j)]$.

输出:G 的一条哈密顿圈 C.

步 1 求图 G 的最小生成树 T.

步 2 设 T 中奇度顶点集合 V',求 V' 的导出子图的最小权完美匹配 M.

步 3 把 M 加到 T 上得到一个欧拉图,求这个欧拉图的欧拉迹．

步 4 沿欧拉迹跳过已经经过的顶点到下一个尚未经过的顶点,得到 G 的哈密顿圈 C.

MM 算法的主要运算量是求最小权完美匹配,时间复杂度为 $O(n^3)$.

定理 4 对于满足三角不等式的流动推销员问题的所有实例 I,有

$$\mathrm{MM}(I) < \frac{3}{2}\mathrm{OPT}(I),$$

并且对于任意的 $\epsilon > 0$,存在无穷多的实例 I 使得

$$\mathrm{MM}(I) \geq \left(\frac{3}{2}-\epsilon\right)\mathrm{OPT}(I).$$

2.3　装箱问题的近似算法

2.3.1　FF 算法与 BF 算法

1.3 节例 7 给出了装箱问题的 FF 算法，这是一个快速近似算法，时间复杂度为 $O(n^2)$，其中 n 是物品数．下述定理给出 FF 算法的近似性能．

定理 5　对于装箱问题的所有实例 I，有

$$\mathrm{FF}(I) \leqslant \frac{17}{10}\mathrm{OPT}(I) + 2,$$

并且存在 OPT(I)任意大的实例 I，使得

$$\mathrm{FF}(I) \geqslant \frac{17}{10}(\mathrm{OPT}(I) - 1).$$

FF 算法关于 1.3 节例 7 中实例的性能比是 5/3，已经接近定理 5 中给出的最坏情况．

对 FF 算法做下述修改得到 **BF 算法**(Best Fit)：在第 i 步把物品 u_i 装进诸装得进 u_i 的箱子中剩余空间最小的箱子，即设在第 $i-1$ 步结束时，箱子 B_k 的剩余空间为 r_k，B_j 满足条件

$$r_j = \min\{r_k \mid r_k \geqslant s_i\}.$$

则在第 i 步将物品 u_i 装进箱子 B_j.

直觉上，BF 算法的近似性能应该比 FF 算法好．但遗憾的是，两者最坏情况的近似性能没有本质区别．

2.3.2　NF 算法

NF 算法(Next Fit)比 FF 算法更简单．它的规则是一只一只地使用箱子，把物品顺序装入箱子．只要箱子装不下要装的物品(不管是否能装下其余未装的物品)就把这只箱子封好，不再使用，而另用一只新的箱子．

定理 6　对于装箱问题的所有实例 I，有

$$\mathrm{NF}(I) \leqslant 2\mathrm{OPT}(I) - 1,$$

并且存在 OPT(I)任意大的实例 I 使得等号成立．

NF 算法的近似性能不如 FF 算法．它的优点是适合于流水线作业，物品随到随装，装好一箱运走一箱．而不是等物品到齐，全部装好后一齐运走．这是一种在线算法(online algorithm)．NF 算法的时间复杂度为 $O(n)$.

2.3.3　FFD 算法和 BFD 算法

分析 1.3 节例 7 中的实例发现，当小物品排在前面，大物品排在后面时，FF 算法的近似性能不好．日常生活的经验也告诉我们，应该先装大物品，然后用小物品填充剩余空隙，这样能更充分地利用空间．根据这个思想，对 FF 算法修改如下：

先将物品按体积从大到小排列，然后采用 FF 算法．这就是 FFD(first fit decreasing)算法．FFD 算法比 FF 算法多一个排序过程，增加计算量 $O(n\operatorname{lb}n)$，时间复杂度仍为 $O(n^2)$.

定理 7 对于装箱问题的所有实例 I，有

$$\mathrm{FFD}(I) \leqslant \frac{11}{9}\mathrm{OPT}(I) + 1,$$

并且存在 OPT(I)任意大的实例 I，使

$$\mathrm{FFD}(I) = \frac{11}{9}\mathrm{OPT}(I).$$

这个结果是越民义于 1991 年给出的.

下述实例给出定理中最坏情况的性能比．

实例 I：有 $30m$ 件物品，

$$s_i = \begin{cases} \dfrac{1}{2} + \varepsilon, & 1 \leqslant i \leqslant 6m, \\ \dfrac{1}{4} + 2\varepsilon, & 6m < i \leqslant 12m, \\ \dfrac{1}{4} + \varepsilon, & 12m < i \leqslant 18m, \\ \dfrac{1}{4} - 2\varepsilon, & 18m < i \leqslant 30m. \end{cases}$$

最优装箱方案和 FFD 算法给出的方案分别如图 2-2(a)和(b)所示，$\mathrm{OPT}(I) = 9m$，$\mathrm{FFD}(I) = 11m$.

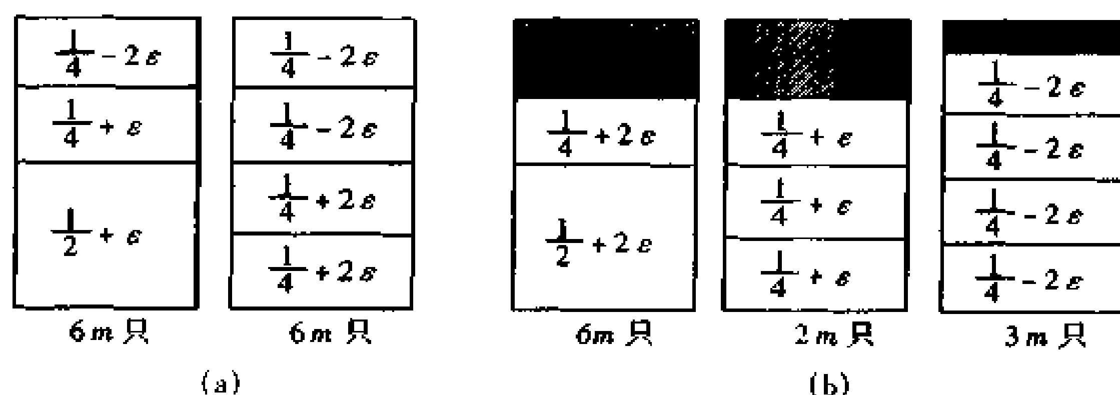


图 2-2

把 FFD 算法中采用 FF 算法的部分改为采用 BF 算法，就得到 BFD 算法(best fit decreasing)．BFD 算法和 FFD 算法最坏情况的近似性能也没有本质的区别．

2.4　0-1 背包问题的近似算法

2.4.1　贪婪算法

0-1 背包问题(0-1 knapsack problem)：给定 n 件物品和一个背包，背包的容量为 b，物品 j 的体积为 a_j，价值为 $c_j(j=1,2,\cdots,n)$. 求使得装入背包的物品的价值之和最大的装法，即求 $J^* \subseteq \{1,2,\cdots,n\}$ 使得

$$\sum_{j\in J^*} c_j = \max\{\sum_{j\in J} c_j \mid \sum_{j\in J} a_j \leqslant b, J \subseteq \{1,2,\cdots,n\}\}.$$

这里 a_j、c_j 以及 b 都是正整数，且 $a_j \leqslant b(j=1,2,\cdots,n)$.

0-1 背包问题可表成下述 0-1 线性规划：

$$\begin{cases} \max \sum_{j=1}^{n} c_j x_j, \\ \text{s.t.} \sum_{j=1}^{n} a_j x_j \leqslant b, \\ \quad x_j \in \{0,1\}, j=1,2,\cdots,n. \end{cases}$$

算法 GA：

步 1　按单位体积的价值从大到小排列物品，不妨设 $c_1/a_1 \geqslant c_2/a_2 \geqslant \cdots \geqslant c_n/a_n$.

步 2　顺序检查每一件物品，如果能装得下就将它装入背包，如果装不下则弃之. 设装入背包的物品总价值为 V.

步 3　设 $c_k = \max\{c_j \mid j=1,2,\cdots,n\}$. 若 $c_k > V$，则将背包内的物品换成物品 k.

定理 8　对于0-1背包问题的所有实例 I，有

$$\mathrm{OPT}(I) < 2\mathrm{GA}(I),$$

并且对于任意的 $\varepsilon > 0$，存在无穷多个实例 I 使得

$$\mathrm{OPT}(I) \geqslant (2-\varepsilon)\mathrm{OPT}(I).$$

2.4.2　多项式时间近似方案

桑尼于 1975 年提出一个算法，把贪婪算法嵌入一个较复杂的过程，得到0-1背包问题的 PTAS.

算法 SA(桑尼，1975)：

输入：0-1 背包问题的实例 I 和 $\varepsilon > 0$.

步 1　令 $m = \lceil 1/\varepsilon \rceil$.

步 2　按单位体积的价值从大到小排列物品，不妨设 $c_1/a_1 \geqslant c_2/a_3 \geqslant \cdots \geqslant c_n/a_n$.

步 3　对每一个 $J \subseteq \{1,2,\cdots,n\}$ 且 $|J| \leqslant m$，若 $\sum_{j\in J} a_j \leqslant b$(否则抛弃这个 J)，

则把 J 中物品装入背包，然后用贪婪算法把剩余物品尽可能多地继续装入背包，即接着顺序检查剩余的物品，只要能装得下就把它装进背包．

步 4　比较得到的所有装法，取其价值最大的作为近似解．

定理 9　对于每一个 $\epsilon > 0$ 和 0-1 背包问题的所有实例 I，有

$$\mathrm{OPT}(I) < (1+\epsilon)\mathrm{SA}(I),$$

算法 SA 的时间复杂度为 $O(n^{2+1/\epsilon})$．

上述定理表明算法 SA 是一个 PTAS.

2.4.3　伪多项式时间算法与完全多项式时间近似方案

令

$$\mathrm{TABLE}(0) = \{(\varnothing, 0, 0)\},$$

$$\begin{aligned}\mathrm{TABLE}(k) = {} & \mathrm{TABLE}(k-1) \\ & \cup \{(J \cup \{k\}, A + a_k, C + c_k) \mid (J, A, C) \\ & \in \mathrm{TABLE}(k-1) \text{ 且 } A + a_k \leqslant b\}, \\ & k = 1, 2, \cdots, n.\end{aligned}$$

显然，TABLE(n) 枚举所有的可行解 J 及其对应的体积 A 和价值 C，其中最大的 C 所对应的 J 即为最优解．

在 TABLE(k) 中若有两个 (J_1, A_1, C_1) 和 (J_2, A_2, C_2) 使得 $A_1 \leqslant A_2$ 且 $C_1 \geqslant C_2$，则继续扩充 J_2 得到的解不可能比继续扩充 J_1 能得到的最好解更好，从而可以从 TABLE(k) 中删去 (J_2, A_2, C_2)．

根据上述递推关系，有下述 0-1 背包问题的动态规划算法．

算法 DP:

步 1　令 $\mathrm{TABLE}(0) = \{(\varnothing, 0, 0)\}$，

步 2　for $k = 1, 2, \cdots, n$ do

(2-1) 令 $M \leftarrow \{(J \cup \{k\}, A + a_k, C + c_k) \mid (J, A, C) \in \mathrm{TABLE}(k-1)$ 且 $A + a_k \leqslant b\}$．

(2-2) 把 M 和 TABLE($k-1$) 归并成 TABLE(k)．若存在 (J_1, A_1, C_1) 和 (J_2, A_2, C_2) 使 $A_1 \leqslant A_2$ 且 $C_1 \geqslant C_2$，则在归并时删去 (J_2, A_2, C_2)．

步 3　找到 $(J^*, A^*, C^*) \in \mathrm{TABLE}(n)$ 使得

$$C^* = \max\{C \mid (J, A, C) \in \mathrm{TABLE}(n)\},$$

则 J^* 是最优解，对应的价值为 C^*．

算法 DP 是 0-1 背包问题的最优化算法，其时间复杂度为 $O(nC^*) = O(n^2 c_{\max})$，其中 $C^* = \mathrm{OPT}(I)$，$c_{\max} = \max\{c_j \mid j = 1, 2, \cdots, n\}$．

需要注意的是，由于正整数 x 在输入中（以二进制形式表示）的长度是 $\lceil \mathrm{lb}(x+1) \rceil$，算法 DP 是指数时间的．但是，如果限制实例中的数（a_j, c_j 以及 b）不超过 n 的某个多项式 $p(n)$，则算法 DP 的时间复杂度为 $O(n^3 p(n))$，成为多项式时间的．这类算法称作**伪多项式时间算法**（pseudo polynomial-time algorithm）．

伊巴拉（O. H. Ibarra）和基姆（C. E. Kim）于 1975 年提出一种"舍入技术"，把伪多项式时间算法改造成 FPTAS，其基本思想是将物品的价值 c_j 缩小若干倍，从而提

高算法DP的效率,其代价是损失一定的精度,最后得到的不是最优解,而是一个近似解.

算法 IK(伊巴拉和基姆,1975):

输入:0-1背包问题的实例 I 和 $\epsilon > 0$.

步1 令 $\alpha \leftarrow \max\{\lceil c_{max}/(1+\frac{1}{\epsilon})n\rceil, 1\}$.

步2 令 $c'_j \leftarrow \lceil c_j/\alpha \rceil, j = 1,2,\cdots,n$. 把以 c'_j 代替 c_j 所得到的实例记作 I'.

步3 对实例 I' 运用算法 DP,求得解 J, J 即为算法求得的近似解.

定理10 对于每一个 $\epsilon > 0$ 和0-1背包问题的所有实例 I,有

$$\text{OPT}(I) < (1+\epsilon)\text{IK}(I),$$

算法IK的时间复杂度为 $O(n^3(1+1/\epsilon))$.

上述定理表明算法IK是0-1背包问题的FPTAS. 兰拉(E. L. Lanlar)于1977年进一步给出0-1背包问题的时间复杂度为 $O(n\text{lb}(1/\epsilon)+1/\epsilon^4)$、空间复杂度为 $O(n+1/\epsilon^3)$ 的FPTAS.

事实上,至今为止所有FPTAS都是用这种舍入技术从伪多项式时间算法得到的.但是,有伪多项式时间算法不一定有FPTAS. 下一节的(0-1)k背包问题($k \geq 2$)就是这样的例子.

2.5 0-1多背包问题的近似算法

0-1多背包问题是0-1背包问题的自然推广:给定 n 件物品和 k 个背包,背包 i 的容量为 $b_i(i = 1,2,\cdots,k)$,物品 j 的体积为 a_j,价值为 $c_j(j = 1,2,\cdots,n)$. 求使得装入 k 个背包的物品的总价值最大的装法. 即求 k 个不相交的集合 $J_i \subseteq \{1,2,\cdots,n\}(i = 1,2,\cdots,k)$ 使得

$$\sum_{j \in J_i} a_j \leq b_i, \quad i = 1,2,\cdots,k$$

且

$$\sum_{i=1}^{k}\sum_{j \in J_i} c_j \text{ 最大}.$$

这里诸 a_j、c_j、b_i 均是正整数.

当 k 为固定的正整数时,把这个问题称作**(0-1)k背包问题**. (0-1)1背包问题就是0-1背包问题.

2.5.1 (0-1)k 背包问题的 PTAS

0-1背包问题的算法SA可以推广到0-1k背包问题上.

算法 MSA:

输入:(0-1)k背包问题的实例 I 和 $\epsilon > 0$.

步1 令 $m \leftarrow \lceil k/\epsilon \rceil$.

步2 按单位体积的价值从大到小排列物品. 不妨设 $c_1/a_1 \geq c_2/a_2 \geq \cdots \geq c_n/a_n$.

步3 对每一种满足条件

$$\sum_{i=1}^{k} |J_i| \leqslant m \text{ 和 } \sum_{j \in J_i} a_j \leqslant b_i, \quad i = 1,2,\cdots,k$$

的 k 个不相交的子集 $J_i \subseteq \{1,2,\cdots,n\}(i = 1,2,\cdots,k)$,以 $J_1, J_2, \cdots, J_k$ 作为初值,用贪婪算法按照排定的顺序将剩余物品继续尽可能地装入背包,即顺序检查每一件剩余物品,只要能装进某个背包,就把它装进这个背包.

步4 比较得到的所有装法,取其价值最大的装法作为近似解.

定理 11 对于每一个正整数 k,算法 MSA 的时间复杂度为 $O(ke^k n^{2+k/\epsilon})$,并且对于每一个 $\epsilon > 0$ 和(0-1)k 背包问题的所有实例 I,有

$$\mathrm{OPT}(I) \leqslant (1+\epsilon)\mathrm{MSA}(I).$$

上述定理表明,对于每一个正整数 k,算法 MSA 是(0-1)k 背包问题的 PTAS.

2.5.2 0-1 多背包问题的 2- 多项式时间近似算法

张立昂等给出 0-1 多背包问题的一个 2- 多项式时间近似算法.算法的主要思想是按单位体积的价值从大到小的顺序检查每一件物品,对于每一个背包 i,先把能装进它的物品放在一起,当这些物品的体积之和不小于 $b_i/2$ 时才把它们装进背包 i.

算法 ZLH(张立昂等,1996 年):

步1 按单位体积的价值从大到小排列物品.不妨设 $c_1/a_1 \geqslant c_2/a_2 \geqslant \cdots \geqslant c_n/a_n$.

按容量从小到大排列背包.不妨设 $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_k$.

步2 令 $M \leftarrow \{1,2,\cdots,k\}, J'_i \leftarrow \varnothing, A'_i \leftarrow 0(i = 1,2,\cdots,k)$.

步3 对 $j = 1,2,\cdots,n$,找到 M 中使 $a_j \leqslant b_i$ 的最小的 i(如果没有这样的 i,则舍去物品 j).

(3-1) 若 $a_j + A'_i < b_i/2$,则令 $A'_i \leftarrow A'_i + a_j, J'_i \leftarrow J'_i \cup \{j\}$.

(3-2) 若 $b_i/2 \leqslant a_j + A'_i \leqslant b_i$,,则令 $J_i \leftarrow J'_i \cup \{j\}, M \leftarrow M - \{i\}$.

(3-3) 若 $a_j + A'_i > b_i$,且存在 $t \in M$ 使 $t > i$,设 t 是其中最小的,则令 $J_i \leftarrow \{j\}, M \leftarrow M - \{i\}$,且

(3-3-1) 若 $A'_i + A'_t < b_t/2$,则令 $A'_t \leftarrow A'_i + A'_t, J'_t \leftarrow J'_i \cup J'_t$.

(3-3-2) 否则($b_t/2 \leqslant A'_i + A'_t \leqslant b_t$).令 $J_t \leftarrow J'_i \cup J'_t, M \leftarrow M - \{t\}$.

(3-4) 否则($a_j + A'_i > b_i$ 且不存在 $t \in M$ 使 $t > i$),令 $M \leftarrow M - \{i\}$ 且若 $c_j \leqslant \sum_{l \in J'_i} c_l$,则令 $J_i \leftarrow J'_i$,否则令 $J_i \leftarrow \{j\}$.

步4 若 $M \neq \varnothing$,则对所有的 $i \in M$,令 $J_i \leftarrow J'_i$.

定理 12 对于0-1多背包问题的所有实例 I,有

$$\mathrm{OPT}(I) \leqslant 2\mathrm{ZLH}(I),$$

并且对于任意的 $\epsilon > 0$ 存在无穷多个实例 I,使得

$$\mathrm{OPT}(I) > (2-\epsilon)\mathrm{ZLH}(I).$$

3　NP 难问题的可近似性

3.1　不可近似性

一个 NP 难的组合最优化问题,能有什么样的性能比的多项式时间近似算法是问题本身的固有性质．由于“P = NP?”和结构复杂性理论中一系列类似问题尚未解决,这类研究只好在某种假设下进行．最常用的当然是假设 P $\neq$ NP.

定理 1　假设 P $\neq$ NP,则

1° 对于任意的常数 $r < 3/2$,装箱问题不存在 r- 多项式时间近似算法．从而,装箱问题不存在 PTAS.

2° 对于每一个 $k \geqslant 2$,(0-1)k 背包问题不存在 FPTAS.

3° (0-1) 多背包问题不存在 6/5- 多项式时间近似算法、从而,(0-1) 多背包问题不存在 PTAS.

4° 对于任意的常数 r,流动推销员问题不存在 r- 多项式时间近似算法．

5° 满足三角不等式的流动推销员问题不存在 PTAS,即存在 $r > 1$ 使得该问题不存在 r- 多项式时间近似算法．

容易设计出(0-1)k 背包问题的伪多项式时间算法．因而,当 $k \geqslant 2$ 时,(0-1)k 背包问题有伪多项式时间算法,有 PTAS,但是没有 FPTAS(除非 P = NP).

3.2　可近似性分类

在 P $\neq$ NP 的假设下,NP 难的组合最优化问题按其可近似性分为 4 类:

(1) 有 FPTAS.

(2) 有 PTAS,但不存在 FPTAS.

(3) 可近似到常数范围内,即存在 r- 多项式时间近似算法,其中 $r > 1$ 是一个常数．但是,不存在 PTAS.

(4) 不可近似到常数范围内,即对任意的常数 $r > 1$,不存在 r- 多项式时间近似算法．

根据上一章的有关定理和 3.1 节中的定理 1,0-1 背包问题属于类(1),(0-1)k 背包问题($k \geqslant 2$)属于类(2),装箱问题、0-1 多背包问题、满足三角不等式的流动推销员问题,属于类(3),流动推销员问题属于类(4).

3.3　问题的可近似性一览表

下述否定结论在 P $\neq$ NP 或者某个类似假设下成立．对于这些假设,人们普遍认为它们成立．

1. 最小点覆盖集

给定无向图 $G=(V,E)$,求 G 的最小点覆盖集,即求 $V'\subseteq V$ 使得 $|V'|$ 最小并且对于每一条边 $\{u,v\}\in E$, u 和 v 至少有一个属于 V'.

该问题可近似到 $2-\mathrm{lblb}|V|/2\mathrm{lb}|V|$ 范围内,但不存在 PTAS.

2. 最小点支配集

给定无向图 $G=(V,E)$,求 G 的最小点支配集,即求 $V'\subseteq V$,使得 $|V'|$ 最小并且对于每一个顶点 $u\in V-V'$,存在 $v\in V'$ 使得 $\{u,v\}\in E$.

该问题可近似到 $1+\mathrm{lb}|V|$ 范围内,但不可近似到常数范围内.

3. 最小边支配集

给定无向图 $G=(V,E)$,求 G 的最小边支配集,即求 $E'\subseteq E$,使得 $|E'|$ 最小并且对于每一条边 $e\in E-E'$,存在 $e'\in E'$ 使得 e 和 e' 相邻.

该问题可近似到 2 范围内.

4. 最小独立支配集

给定无向图 $G=(V,G)$,求 G 的最小独立支配集,即求 $V'\subseteq V$,使得 $|V'|$ 最小并且满足下述条件:① 对于每一个顶点 $u\in V-V'$,存在 $v\in V'$ 使得 $\{u,v\}\in E$,② V' 的任意两点之间没有边.

对于任意的 $\varepsilon>0$,该问题不可近似到 $|V|^{1-\varepsilon}$ 范围内.

5. 图的着色

给定无向图 $G=(V,E)$, G 的 k 着色是把 V 划分成 k 个互不相交的子集 $V_1,V_2,\cdots,V_k$,使得每一个 V_i 中的任意两点之间都没有边. k 称作着色的色数. 求 G 的色数最小的着色.

该问题可近似到 $O(|V|(\mathrm{lblb}|V|)^2/(\mathrm{lb}|V|)^3)$ 范围内,并且对任意的 $\varepsilon>0$,不可近似到 $|V|^{1/5-\varepsilon}$ 范围内.

6. 图的边着色

给定无向图 $G=(V,E)$, G 的 k 边着色是把 E 划分成 k 个互不相交的子集 $E_1,E_2,\cdots,E_k$,使得每一个 E_i 中的任意两条边都不相邻. k 称作边着色的色数. 求 G 的色数最小的边着色.

该问题可近似到 4/3 范围内. 并且对于任意的 $\varepsilon>0$,不可近似到 $4/3-\varepsilon$ 范围内,从而,不存在 PTAS.

7. 最大团

给定无向图 $G=(V,E)$,求 G 的最大团,即求 $V'\subseteq V$,使得 $|V'|$ 最大并且 V' 中的任意两点之间都有边.

该问题可近似到 $O(|V|/(\mathrm{lb}|V|)^2)$,并且对任意的 $\varepsilon>0$,不可近似到 $|V|^{1-\varepsilon}$ 范围内.

8. 最大独立集

给定无向图 $G=(V,E)$,求 G 的最大独立集,即求 $V'\subseteq V$,使得 $|V'|$ 最大并且 V' 中的任意两点之间没有边.

由于求 G 的最大独立集等同于求 G 的补图的最大团,因此该问题与最大团问题有相同的可近似性.

9. 最大割集

给定无向图 $G=(V,E)$,求 G 的最大割集,即求 $V'\subseteq V$,使得 $|(V',V-V')|$ 最大,其中

$$(V',V-V')=\{\{u,v\}\in E \mid u\in V' 且 v\in V-V'\}.$$

该问题可近似到2范围内,存在平均性能比为1.139的多项式时间随机近似算法,但不存在PTAS.

10. 流动推销员问题

给定 n 个城市 $c_1,c_2,\cdots,c_n$ 及每一对城市 c_i 和 c_j 之间的距离 $d(i,j)\in \mathbf{Z}^+$,这里 $d(i,j)=d(j,i)$. 求长度最短的环游,即求 $1,2,\cdots,n$ 的排列 π 使得

$$\min\left(\sum_{i=1}^{n-1} d(\pi(i),\pi(i+1))+d(\pi(n),\pi(1))\right).$$

存在 $\delta>0$ 使得该问题不可近似到 $2^{n^{\delta}}$ 范围内.

11. 满足三角不等式的流动推销员问题

假设在流动推销员问题中距离满足三角不等式,即对任意的3个城市 c_i、c_j、c_k,

$$d(i,k)+d(k,j)\geqslant d(i,j).$$

该问题可近似到3/2范围内,但不存在PTAS.

12. 混合图上的中国邮路问题

给定混合图 $G=(V,A,E)$ 及每一条边或弧 $e\in A\cup E$ 的长度 $l(e)\in \mathbf{Z}^+$,求 G 中经过每一条边和弧至少一次的长度最短的复杂回路.

该问题可近似到5/3范围内.

13. 最长路径

给定无向图 $G=(V,E)$,求 G 中最长的简单路径,即求不同的顶点序列 $v_1,v_2,\cdots,v_m$,使得对于所有的 $1\leqslant j\leqslant m-1$, $\{v_j,v_{j+1}\}\in E$,并且 m 最大.

该问题可近似到 $O(|V|/\text{lb}|V|)$ 范围内,但是不可近似到常数范围内.

14. 装箱问题

给定有穷的物品集合 U、每一件物品 $u\in U$ 的体积 $s(u)\in(0,1]$,箱子的容积为1. 把 U 划分成 k 个互不相交的子集 $U_1,U_2,\cdots,U_k$,使得对于每一个 $1\leqslant i\leqslant k$, $\sum_{u\in U_i}s(u)\leqslant 1$,并且 k 最小.

该问题可近似到3/2范围内和71/60 + 78/71OPT(I)范围内,但是对于任意的 $\varepsilon>0$,不可近似到 $3/2-\varepsilon$ 范围内.

15. 最短公共超字符串

给定字母表 Σ 上有穷的字符串集合 R,求最短的字符串 $w\in\Sigma^*$,使得每一个 $x\in R$ 都是 w 的子串,即存在 u、$v\in\Sigma^*$,使得 $w=uxv$.

该问题可近似到2.75范围内,但不存在PTAS.

16. 限制拖延的排序

给定有穷的任务集 T,每一项任务 $t\in T$ 的执行时间 $l(t)\in\mathbf{Z}^+$,权 $w(t)\in\mathbf{Z}^+$

和限期 $d(t)\in \mathbf{Z}^+$，对子集 $S\subseteq T$ 以及正整数 K，T 的单处理机时间表是一个映射 $\sigma: T\to \mathbf{Z}_0^+$ 且满足下述条件：对所有的 t_1、$t'\in T$，或者 $\sigma(t)+l(t)\leqslant\sigma(t')$，或者 $\sigma(t')+l(t')\leqslant\sigma(t)$. 求 T 的单处理机时间表 σ，使得①所有拖延的任务 t(即 $\sigma(t)+l(t)>d(t)$)的权 $w(t)$之和不超过 K，②S 中按期完工的任务 t(即 $\sigma(t)+l(t)\leqslant d(t)$)的个数最多．其中 $\mathbf{Z}_0^+$ 是非负整数集．

存在 $\delta>0$ 使得该问题不可近似到 $|T|^\delta$ 范围内．

17．带优先约束和延迟的排序

给定有穷的任务集 T，T 上的偏序关系 $\prec$、正整数 D 以及每一件任务 $t\in T$ 的延迟 $d(t)(0\leqslant d(t)\leqslant D)$. 求 T 的单处理机时间表 $\sigma: T\to\mathbf{Z}_0^+$，使得①对任意两项任务 t_1、$t_2\in T$，若 $t_1\prec t_2$，则 $\sigma(t_2)-\sigma(t_1)>d(t_1)$，②总完工时间最早，即 $\max\{\sigma(t)\mid t\in T\}$ 最小．

该问题可近似到 $2-1/(D+1)$ 范围内．

18．多处理机时间表

给定有穷的任务集 T 和 m 台机器，每一项任务在一台机器上加工完成，任务 $t\in T$ 在机器 i 上的加工时间为 $l(t,i)$. 求映射 $f: T\to\{1,2,\cdots,m\}$，使得总完工时间最早，即

$$\max_{1\leqslant i\leqslant m}\sum_{f(t)=i} l(t,i)$$

最小．

该问题可近似到 2 范围内，但不存在 PTAS. 当机器数 m 固定时，问题有 FPTAS. 当 $l(t,i)$ 与 i 无关(即所有机器相同)时，问题有 PTAS.

19．有优先约束的多处理机时间表

给定有穷的任务集 T、m 台相同的机器和 T 上的偏序关系 $\prec$，每一项任务在一台机器上加工完成，加工时间均为 1. 求 T 的时间表 $f: T\to\mathbf{Z}_0^+$，使得①满足 T 上的优先约束，即对任意两项任务 t_1、$t_2\in T$，若 $t_1\prec t_2$，则 $f(t_1)<f(t_2)$，②对每一时刻 $u\geqslant 0$，$|\{t\in T\mid f(t)=u\}|\leqslant m$，③总完工时间最早，即 $\max\{f(t)\mid t\in T\}$ 最小．

该问题可近似到 $2-2/|T|$ 范围内，但不存在 PTAS.

20．有资源限制的多处理机时间表

给定有穷的任务集 T，m 台相同的机器和 s 种资源；资源 i 可供使用的上限为 $b_i(1\leqslant i\leqslant s)$；每一项任务 $t\in T$ 在一台机器上加工完成，加工时间为 $l(t)$，占用的资源 i 为 $r_i(t)$，$0\leqslant r_i(t)\leqslant b_i$，$1\leqslant i\leqslant s$. 求 T 的时间表 $f: T\to\mathbf{Z}_0^+$，使得①至多同时使用 m 台机器，即对每一时刻 $u\geqslant 0$，$|S(u)|\leqslant m$，其中

$$S(u)=\{t\in T\mid f(t)\leqslant u<f(t)+l(t)\},$$

②服从资源限制，即对每一时刻 $u\geqslant 0$ 和每一种资源 $i(1\leqslant i\leqslant s)$，$\sum_{t\in S(u)} r_i(t)\leqslant b_i$，③总完工时间最早，即

$$\max\{f(t)+l(t)\mid t\in T\}$$

最小．

该问题可近似到2范围的．

21. 自由作业时间表

给定有穷的作业集 J 和 m 台机器，每一个作业 $j \in J$ 由 m 道工序 o_{ij} 组成($1 \leq i \leq m$)．o_{ij} 由机器 i 完成，加工时间为 $l_{ij} \in \mathbf{Z}_0^+$；$J$ 的自由作业时间表是 m 个单处理机时间表 $f_i: J \to \mathbf{Z}_0^+$，$1 \leq i \leq m$，且满足下述条件：① 对任意两个 j、$j' \in J$ 和每一个 $i(1 \leq i \leq m)$，$f_i(j) < f_i(j')$ 蕴涵 $f_i(j) + l_{ij} \leq f_i(j')$，② 对每一个 $j \in J$，所有区间$[f_i(j), f_i(j) + l_{ij})(1 \leq i \leq m)$ 互不相交．求总完工时间最早的自由作业时间表，即使得

$$\max\{f_i(j) + l_{ij} \mid 1 \leq i \leq m, j \in J\}$$

最小．

该问题可近似到2范围内，并且对任意的 $\varepsilon > 0$，不可近似到 $5/4 - \varepsilon$ 范围内，从而不存在 PTAS.

22. 流水作业时间表

给定有穷的作业集 J 和 m 台机器，每一项作业 $j \in J$ 由 m 道工序 o_{ij} 组成($1 \leq i \leq m$)．o_{ij} 由机器 i 完成，加工时间为 $l_{ij} \in \mathbf{Z}_0^+$；$J$ 的流水作业时间表是一个自由作业时间表 $f_i: J \to \mathbf{Z}_0^+$，$1 \leq i \leq m$，且满足条件 ③ 对每一个 $j \in J$ 和 $i(1 \leq i \leq m - 1)$，$f_i(j) + l_{ij} \leq f_{i+1}(j)$．求总完工时间最早的流水作业时间表，即使得

$$\max\{f_i(j) + l_{ij} \mid 1 \leq i \leq m, j \in J\}$$

最小．

该问题当 m 为偶数时可近似到$m/2$范围内，当 m 为奇数时可近似到$m/2 + 1/6$范围内，并且对任意的 $\epsilon > 0$，不可近似到 $5/4 - \epsilon$ 范围内．

23. 作业时间表

给定有穷的作业集 J 和 m 台机器，每一个作业 $j \in J$ 由 n_j 道工序 o_{ij} 组成($1 \leq i \leq n_j$)．工序 o_{ij} 由机器 $p_{ij} \in \{1, 2, \cdots, m\}$ 完成，加工时间为 $l_{ij} \in \mathbf{Z}_0^+$；记

$$O_p = \{o_{ij} \mid p_{ij} = p\}, \quad 1 \leq p \leq m,$$

J 的作业时间表是 m 个单处理机时间表$f_p: O_p \to \mathbf{Z}_0^+$，$1 \leq p \leq m$，且满足条件 ④ 对所有 $o_{ij} \in O_p$ 且 $1 \leq i \leq n_j - 1$，$f_p(o_{ij}) + l_{ij} \leq f_p(o_{i+1,j})$．求 J 的总完工时间最早的作业时间表，即使得

$$\max\{f_p(o_{ij}) + l_{ij} \mid i = n_j, o_{ij} \in O_p, j \in J, 1 \leq p \leq m\}$$

最小．

该问题可近似到 $O(\mathrm{lb}^2(m\mu)/\mathrm{lb}\,lb(m\mu))$，其中 $\mu = \max\{n_j \mid j \in J\}$，并且对任意的 $\epsilon > 0$，不可近似到 $5/4 - \epsilon$．

24. 0-1 背包问题

给定 n 件物品的体积$a_j \in \mathbf{Z}^+$，价值 $c_j \in \mathbf{Z}^+$ $(1 \leq j \leq n)$ 以及背包的容量 $b \in \mathbf{Z}^+$．求 $J \subseteq \{1, 2, \cdots, n\}$，使得

$$\sum_{j \in J} a_j \leq b,$$

且

$$\sum_{j \in J} c_j \text{ 最大}.$$

该问题是有时间复杂度 $O(n\mathrm{lb}(1/\varepsilon)+1/\varepsilon^4)$ 及空间复杂度 $O(n+1/\varepsilon^3)$ 的 FPTAS.

25. 整数背包问题

给定正整数 a_j、$c_j(j=1,2,\cdots,n)$ 以及 b,求非负整数 $x_1,x_2,\cdots,x_n$ 使得

$$\sum_{j=1}^{n} a_j x_j \leqslant b,$$

且

$$\sum_{j=1}^{n} c_j x_j \text{ 最大}.$$

该问题是有时间复杂度和空间复杂度为 $O(n+1/\varepsilon^3)$ 的 FPTAS.

26. 0-1 多背包问题

给定 n 件物品的体积$a_j\in\mathbf{Z}^+$,价值 $c_j\in\mathbf{Z}^+$ $(1\leqslant j\leqslant n)$ 和 k 个背包的容量$b_i\in\mathbf{Z}^+$ $(1\leqslant i\leqslant k)$;求 k 个互不相交的子集 $J_i\subseteq\{1,2,\cdots,n\}$,使得

$$\sum_{j\in J_i} a_j \leqslant b_i,\quad 1\leqslant i\leqslant k,$$

且

$$\sum_{i=1}^{k}\sum_{j\in J_i} c_j \text{ 最大}.$$

该问题可近似到 2 范围内,但不可近似到 6/5 范围内.从而不存在 PTAS.

当背包数 k 固定时,称作(0-1)k 背包问题.对每一个 $k\geqslant 2$,(0-1)k 背包问题有 PTAS,但没有 FPTAS.

27. 0-1 多约束背包问题

给定有穷的物品集 U 和 k 个子集 $U_i\subseteq U$,$1\leqslant i\leqslant k$.每一件物品 $u\in U$ 的体积为 $s(u)\in\mathbf{Z}^+$,价值为 $c(u)\in\mathbf{Z}^+$ 以及 k 个容量限制$b_i\in\mathbf{Z}^+$,$1\leqslant i\leqslant k$,求子集 $U'\subseteq U$ 使得

$$\sum_{u\in U'\cap U_i} s(u)\leqslant b_i,\quad 1\leqslant i\leqslant k,$$

且

$$\sum_{u\in U'} c(u) \text{ 最大}.$$

对任意的 $\varepsilon>0$,该问题不可近似到 $n^{1/2-\varepsilon}$ 范围内,其中 n 是物品数.

当约束数 k 固定时,称作(0-1)k **背包问题**.对于每一个正整数 k,(0-1)k 约束背包问题有 FPTAS.

这两个问题的整数形式(即每一种物品可以装多件)分别称作**整数多约束背包问题**和**整数 k 约束背包问题**.它们的可近似性分别与(0-1) 多约束背包问题和 0-1k 约束背包问题相同.

28. 整数 k 选择背包问题

给定 $n\times k$ 非负整数矩阵$\boldsymbol{A}$ 和$\boldsymbol{C}$,以及 $b\in\mathbf{Z}_0^+$,这里 k 是固定的正整数.求 n 维非负整数向量 $\boldsymbol{x}$ 和映射$f:\{1,2,\cdots,n\}\to\{1,2,\cdots,k\}$,使得

$$\sum_{i=1}^{n} a_{i,f(i)}x_i \leqslant b,$$

且
$$\sum_{i=1}^{n} c_{i,f(i)} x_i \text{ 最大}.$$

该问题有 FPTAS.

29. 最大可满足性(MAX SAT)

给定有穷的变量集 U 和 U 上文字的析取子句的有穷集合 C,这里对每一个变量 $x \in U$,x 和 $\neg x$ 称作文字. 求 U 的真假值赋值,使得 C 中成真子句的个数最多.

该问题可近似到4/3范围内,但不存在 PTAS. 设 $k \geq 2$ 是一个固定的正整数,如果每一个子句恰好是 k 个文字的析取,则问题可近似到 $1/(1-2^{-k})$ 范围内.

4 局部搜索算法

4.1 局部搜索算法的一般描述

局部搜索算法(local search algorithm)是一类近似算法的通称,它从一个初始解开始,每一步在当前解的邻域内找到一个更好的解,使目标函数逐步优化,直至不能进一步改进为止. 局部搜索算法通常得到的是局部最优解. 为了得到全局最优解,需要从多个初始解开始,重复进行搜索. 局部搜索算法灵活、简便. 实验表明,局部搜索算法对很多著名的 NP 难问题,如可满足性问题,取得巨大的成功.

4.1.1 邻域与局部最优解

给定组合最优化问题 Π 的实例 I,设 I 的可行解集为 S. 对每一个可行解 $f \in S$,定义一个子集 $N(f) \subseteq S$,称作 f 的邻域(neighborhood).

如果对于所有的 $g \in N(f)$,
$$m(I,f)\begin{cases} \geq m(I,g), & \text{若 } \Pi \text{ 是最大化问题}, \\ \leq m(I,g), & \text{若 } \Pi \text{ 是最小化问题}, \end{cases}$$
则称 f 是关于邻域 N 的局部最优解. 当不会产生混淆时,简称 f 是**局部最优解**(locally optimal solution). 这里 $m(I,f)$ 是目标函数值.

为了区别于局部最优解,也把问题的最优解称作**全局最优解**(globally optimal solution).

例 1 流动推销员问题的 k 交换邻域 N_k. 给定流动推销员问题的实例 I. 设正整数 $k \geq 2$,f 是 I 的一个环游. 定义
$$N_k(f) = \{g \mid g \text{ 是 } I \text{ 的环游且与 } f \text{ 至多有 } k \text{ 条边不同}\}.$$

N_2 和 N_3 分别如图4-1和图4-2所示,其中(a)是环游 f,(b)是 $N_2(f)$ 或 $N_3(f)$ 中的一条环游.

4.1.2 算法的一般格式

算法 LOCAL SEARCH:

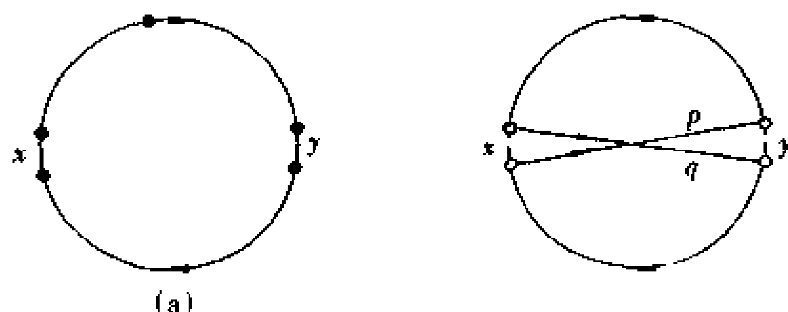


图 4-1

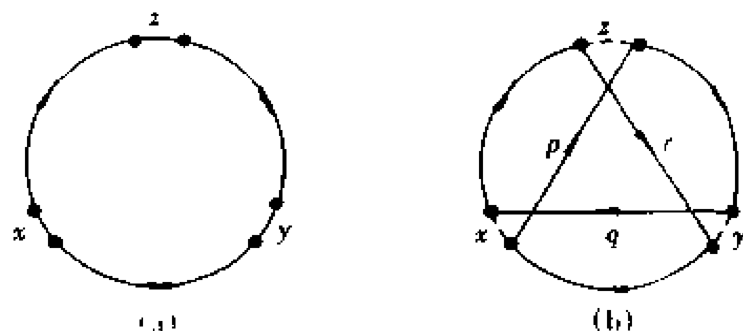


图 4-2

步 1　生成初始近似解 x.

步 2　while improve(x) $\neq$ "no"do

　　　$x \leftarrow$ improve(x).

步 3　return x.

其中 improve(x) 是一个子程序．如果 $N(x)$ 中有比 x 更好的可行解，则 improve(x) 返回一个这样的可行解，否则返回"no".

设计局部搜索算法主要包括：

(1) 如何生成初始近似解 x.

(2) 选择适当的邻域结构．

(3) 设计算法实现 improve(x).

局部搜索算法得到的是局部最优解，局部最优解不一定是全局最优解．为了得到全局最优解，需要多次重复局部搜索，每次从一个新的初始解开始．邻域越大，算法得到的局部最优解是全局最优解的可能性越大，但 improve(x) 的复杂度也随着增大．在设计算法时必须很好地照顾到这两个相互矛盾的方面．设算法得到的局部最优解是全局最优解的概率为 p_0，计算时间为 t_0．给定时间 t，在时间 t 内可重复 $k = \lceil t/t_0 \rceil$ 次，得到 k 个局部最优解．这 k 个局部最优解中有全局最优解的概率为 $1-(1-p_0)^k$．这个概率越大，算法越有效．

设计有效的局部搜索算法是一门艺术，往往靠的是人的智慧和经验，并通过实验进行检验和逐步改进．

4.1.3　选择初始解的策略

设需要 n 个初始解，$n = km, k \geqslant 1$.

1. 常用策略

从可行解 S 中有放回地独立抽取 n 次，每次等可能地抽取一个可行解作为初始解，即在每次开始局部搜索时随机地生成一个可行解作为初始解.

2. 划分策略

把 S 划分成 m 个互不相交的子集 $S_1, S_2, \cdots, S_m$，从每个 S_i 中用常用策略抽取 k 个可行解，共得到 $n = km$ 个作为初始解.

3. 无放回取样策略

从 S 中无放回地抽取 m 次，每次等可能地抽取一个可行解，独立重复进行 k 次，得到 $n = km$ 个初始解.

4.2　流动推销员问题的局部搜索算法

流动推销员问题的邻域采用4.1节例1中的 k 交换邻域 N_k. 关于 N_k 的局部最优解称作 k 最优环游. 林(V.S.Lin)于1965年通过实验发现以 N_3 为最好. 对于他计算的48个城市的问题，3-最优环游是最优环游的概率约为0.05，因此从100个随机初始点开始的实验以0.99的概率得到最优环游. 他估计对于 n 个城市的流动推销员问题，3-最优环游是最优环游的概率 $p(n) \approx 2^{-n/10}$. 给定获得全局最优环游的概率，根据这个估计可以计算出需要重复的试验次数.

设 f 是一条环游，环游 $g \in N_3(f)$，当且仅当 g 可以如下得到：删去 f 中 $l \geqslant 1$ 个连续的城市 $(t_1, t_2, \cdots, t_l)$，然后再将它照原样或反转后重新插入到剩余的某两个相邻城市之间.

设城市集合 $C = \{1, 2, \cdots, n\}$，i 和 j 之间的距离为 $d(i,j) \in \mathbf{Z}^+$，这里 $d(i,j) = d(j,i)$. 设环游 $f = (t_1, t_2, \cdots, t_n)$，$1 \leqslant k < j < n$. 从 f 中删去 $\sigma = (t_1, t_2, \cdots, t_k)$，有两种方式将 σ 重新插入.

(1) 将 σ 照原样插入 t_j 和 t_{j+1} 之间，得到 $g_1 = (t_{j+2}, \cdots, t_n, t_{k+1}, \cdots, t_j, t_1, \cdots, t_k, t_{j+1})$，即从 f 中删去边 $\{t_1, t_n\}$、$\{t_k, t_{k+1}\}$ 和 $\{t_j, t_{j+1}\}$，代之以 $\{t_j, t_1\}$、$\{t_n, t_{k+1}\}$ 和 $\{t_k, t_{j+1}\}$，如图4-3(b)所示.

(2) 将 σ 反转后插入 t_j 和 t_{j+1} 之间，得到 $g_2 = (t_{j+2}, \cdots, t_n, t_{k+1}, \cdots, t_j, t_k, \cdots, t_1, t_{j+1})$，即从 f 中删去边 $\{t_n, t_1\}$、$\{t_k, t_{k+1}\}$ 和 $\{t_j, t_{j+1}\}$，代之以 $\{t_1, t_{j+1}\}$、$\{t_n, t_{k+1}\}$ 和 $\{t_k, t_j\}$. 如图4-3(c)所示. 当 $j = k + 1$ 时，g_2 蜕变成 $g_3 = (t_{j+2}, \cdots, t_n, t_j, \cdots, t_1, t_{j+1})$，即从 f 中删去边 $\{t_1, t_n\}$ 和 $\{t_j, t_{j+1}\}$，代之以 $\{t_1, t_{j+1}\}$ 和 $\{t_n, t_j\}$. 如图4-3(d)所示.

算法　LIN TSP(林，1965)：

输入：$n \times n$ 对称的距离矩阵 $[d(i,j)]$.

输出：环游 $(t_1, t_2, \cdots, t_n)$.

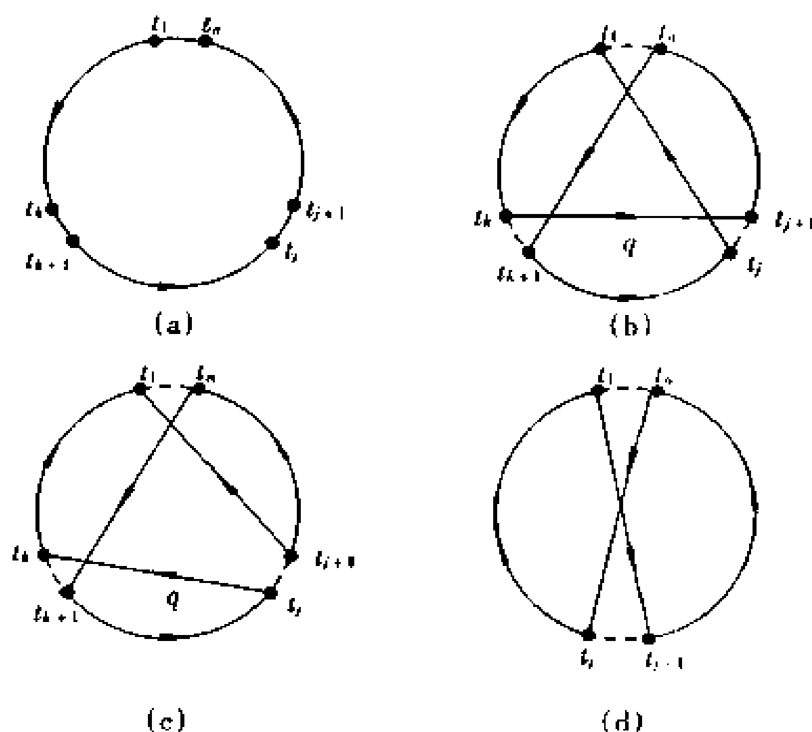


图 4-3

步 1　随机生成环游$(t_1,t_2,\cdots,t_n)$.

步 2　for count $=1$ to n do

(2-1) for $k=1$ to $n-3$ do

for $j=k+1$ to $n-1$ do

(2-1-1) if $d(t_k,t_{j+1})+d(t_1,t_j)\leqslant d(t_1,t_{j+1})+d(t_k,t_j)$

then $d\leftarrow d(t_k,t_{j+1})+d(t_1,t_j)$ and $L\leftarrow 4$

else $d\leftarrow d(t_1,t_{j+1})+d(t_k,t_j)$ and $L\leftarrow 6$.

(2-1-2) if $d+d(t_{k+1},t_n)$(加入的边长) $<d(t_1,t_n)+d(t_k,t_{k+1})+d(t_j,t_{j+1})$(删去的边长) goto L.

(2-2) $(t_1,\cdots,t_n)\leftarrow(t_n,t_1,\cdots,t_{n-1})$.

步 3　return $(t_1,\cdots,t_n)$.

步 4　$(t_1,\cdots,t_n)\leftarrow(t_{j+2},\cdots,t_n,t_{k+1},\cdots,t_j,t_1,\cdots,t_k,t_{j+1})$.

步 5　goto 2.

步 6　$(t_1,\cdots,t_n)\leftarrow(t_{j+2},\cdots,t_n,t_{k+1},\cdots,t_j,t_k,\cdots,t_1,t_{j+1})$.

步 7　goto 2.

4.3　图的均匀划分的局部搜索算法

图的均匀划分问题:给定 $2n$ 个顶点的无向完全图 G,其顶点集 $V=\{1,2,\cdots,2n\}$,$[c(i,j)]$是定义在 $V\times V$上的对称的费用矩阵,其中 $c(i,i)=0,1\leqslant i\leqslant 2n$.

若 $A、B\subseteq V$ 使得 $A\cup B=V, A\cap B=\varnothing$，且 $|A|=|B|=n$，则称 $A、B$ 是 G 的一个均匀划分．划分 $A、B$ 的费用定义为

$$c(A,B)=\sum_{\substack{i\in A\\ j\in B}}c(i,j).$$

求图 G 的费用最小的均匀划分．

设 $A、B$ 是一个均匀划分，$a\in A, b\in B$．令

$$A'=(A-\{a\})\cup\{b\},\quad B'=(B-\{b\})\cup\{a\},$$

则 $A'、B'$ 也是均匀划分．称由 $A、B$ 这样得到 $A'、B'$ 的运算为一个交换．

$a\in A$ 的外费用 $E(a)$ 定义为

$$E(a)=\sum_{i\in B}c(a,i),$$

内费用 $I(a)$ 定义为

$$I(a)=\sum_{j\in A}c(a,j).$$

可类似地定义 $b\in B$ 的外费用 $E(b)$ 和内费用 $I(b)$．

对每一个 $v\in V$，令

$$D(v)=E(v)-I(v),$$

称作 v 的费用差．

交换 $a\in A$ 和 $b\in B$ 引起的费用减少等于

$$g(a,b)=D(a)+D(b)-2c(a,b).$$

g 称作关于 $A、B$ 的**交换增益函数**．

设 $A、B$ 经过交换 $a\in A$ 和 $b\in B$ 得到 $A'、B'$．关于 $A'、B'$ 的费用差 D' 可由关于 $A、B$ 的费用差 D 得到：

$$\begin{aligned}
D'(x)&=D(x)+2c(x,a)-2c(x,b), && x\in A-\{a\},\\
D'(y)&=D(y)+2c(y,b)-2c(y,a), && y\in B-\{b\},\\
D'(z)&=-D(z)+2c(a,b), && z\in\{a,b\}.
\end{aligned}$$

利用交换可以自然地给出问题的邻域．对于每一个均匀划分 $A、B$，**交换邻域** $N_1(A,B)$ 定义为由 $A、B$ 经过一次交换得到的所有均匀划分．

局部搜索算法从随机生成的均匀划分 $A、B$ 开始，检查每一对 $a\in A$ 和 $b\in B$，只要找到一对 $a、b$ 使 $g(a,b)>0$，就交换 a 和 b，得到一个新的均匀划分．修改每一个 $v\in V$ 的费用差 $D(v)$．重复这个过程，直到对每一对 $a\in A$ 和 $b\in B$ 都有 $g(a,b)\leqslant 0$．此时 $A、B$ 为一个局部最优解．

类似流动推销员问题，可以定义图的均匀划分问题的 **k 交换邻域**：至多交换 k 个元素．k 交换邻域的大小为 $O(n^{2k})$．随 k 值的增加，改善均匀划分和判断局部最优化所需的时间迅速增加．

克尼根(B.W.Kernighan)与林于1969年提出一种新的搜索策略，称作**变深度搜索技术**，其思想是用寻找一个好的交换序列来代替寻找一个好的交换，这个好的交换序列的长度是变动的．具体做法如下：从一个随机的初始均匀划分 $A、B$ 开始，找到 $a_1\in A$ 和 $b_1\in B$，使 $g(a_1,b_1)$ 尽可能地大，交换 a_1 和 b_1，修改费用差 D．再找

到 $a_2 \in A - \{a_1\}$ 和 $b_2 \in B - \{b_1\}$，使 $g(a_2, b_2)$ 尽可能地大，交换 a_2 和 b_2，修改费用差 D. 如此重复进行，得到 n 对 $(a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)$，其中 $a_1, a_2, \cdots, a_n \in A$ 且各不相同，$b_1, b_2, \cdots, b_n \in B$ 且各不相同. 对于每一个 $t(1 \leqslant t \leqslant n)$，交换 $a_1, a_2, \cdots, a_t$ 和 $b_1, b_2, \cdots, b_t$ 从 A、B 得到 A_t、B_t，其费用减少

$$G_t = \sum_{i=1}^{t} g_i(a_i, b_i).$$

这里 g_i 是关于 A_{i-1}、B_{i-1} 的交换增益函数，且 $A_0 = A, B_0 = B$. 注意到 $A_n = B, B_n = A$，恒有 $G_n = 0$. 设

$$G_k = \max\{G_t \mid 1 \leqslant t \leqslant n\}.$$

若 $G_k > 0$，则以 A_k、B_k 代替 A、B，然后重复上述过程；若 $G_k \leqslant 0$，则取 A、B 作为近似解.

当 $G_k > 0$ 时，可能有某些 $i(1 \leqslant i < k)$ 使 $g_i(a_i, b_i) \leqslant 0$. 与以交换邻域 N_i 为邻域的普通局部搜索算法比较，此法一步做了 k 个交换，并且允许"平移"(费用不变)和"上升移动"(费用增加). 当 $g_i(a_i, b_i) = 0$ 时交换 a_i 和 b_i，相当于平移；当 $g_i(a_i, b_i) > 0$ 时交换 a_i 和 b_i，相当于上升移动. 而普通局部搜索算法只允许下降移动(使费用减少的交换). 平移和上升移动的作用是移出原有的邻域，以避免过早陷入局部最优解. 此法与以 k 交换邻域为邻域的普通局部搜索算法比较，其一，这里的 k 不是预先给定的常数，而是计算中确定的，且各步的 k 值是不同的，其二，不需要搜索整个的 k 交换邻域.

算法 KL(克尼根与林，1969)：

输入：$2n$ 个顶点的无向完全图 G 的对称的费用矩阵 $[c(i,j)]$.

输出：G 的均匀划分 A、B.

步 1　从 $V = \{1, 2, \cdots, 2n\}$ 中随机地取 n 个点作为 A，剩余的 n 个点作为 B.

步 2　计算每一个 $v \in V$ 关于均匀划分 A、B 的费用差 $D(v)$.

步 3　令 $A_1 \leftarrow A, B_1 \leftarrow B$.

步 4　for $i = 1$ to n do

(4-1) 找到 $a_i \in A_i$ 和 $b_i \in B_i$ 使

$$\begin{aligned} g_i &= D(a_i) + D(b_i) - 2c(a_i, b_i) \\ &= \max\{D(x) + D(y) - 2c(x, y) \mid x \in A_i, y \in B_i\}. \end{aligned}$$

(4-2) 令 $A_{i+1} \leftarrow A_i - \{a_i\}, B_{i+1} \leftarrow B_i - \{b_i\}$.

(4-3) 对所有的 $x \in A_{i+1}$，令 $D(x) \leftarrow D(x) + 2c(x, a_i) - 2c(x, b_i)$.

对所有的 $y \in B_{i+1}$，令 $D(y) \leftarrow D(y) + 2c(y, b_i) - 2c(y, a_i)$.

步 5　求 k 使得

$$G = \sum_{i=1}^{k} g_i = \max\{\sum_{i=1}^{t} g_i \mid 1 \leqslant t \leqslant n\}.$$

步 6　if $G \leqslant 0$ then return A, B.

步 7　记 $X = \{a_1, \cdots, a_k\}, Y = \{b_1, \cdots, b_k\}$.

令 $A \leftarrow (A - X) \cup Y$,　$B \leftarrow (B - Y) \cup X$.

步 8 goto 4.

实验表明这个变深度搜索算法是成功的．据克尼根和林的报告，当 $2n=30$ 时，得到全局最优解的概率大约为0.5，而以 N_1 为邻域的普通局部搜索法的这个概率仅约为 0.1. 他们估计，实际使用中，当顶点数为 $2n$ 时，得到全局最优解的概率 $p(n)\approx 2^{-n/15}$，算法的运行时间为 $O(n^{2.4})$. 他们还把这个技术成功地运用于流动推销员问题. 此时，为了保证边交换序列的可行性，算法的细节要复杂得多．

4.4 n 后问题的局部搜索算法

n 后问题(n-queens problem)：在 $n\times n$ 的棋盘上放置 n 个皇后，要求皇后之间不发生冲突，即任何两个皇后不在同一行、同一列和同一条斜线上．满足上述条件的放置方式是 n 后问题的一个解．

对于每一个 $n\geq 4$，n 后问题都有解，并且存在这样的解. 它的每个皇后的坐标可以用公式给出．不过，这种解是一种很特殊的放置方式．现在的问题是，如何得到 n 后问题的"任意"解．

n 后问题是一个约束满足问题(constraint satisfaction problem)，解这类问题通常采用回溯搜索，计算时间随问题实例的规模指数增长，因而只能适用于规模不大的实例．明顿(S. Minton)等人于 1990 年提出"冲突最小化"思想，把约束满足问题转化成最小化问题，采用冲突最小化的启发式算法求解，n 后问题和时间表问题均取得成功. 他们采用的算法实质上是一种局部搜索算法．索西克(R. Sosic)和顾钧(Jun Gu)成功地设计出 n 后问题的高效局部搜索算法，使用 IBM RS6000，在 55 秒内成功地找到 300 万个皇后问题的解．下面介绍他们的算法．

每一行每一列置放一个皇后，称作一个布局．用 $r[i]$ 表示第 i 列中的皇后所在的行号．每一个布局可表成 $1,2,\cdots,n$ 的一个排列 $r[1],r[2],\cdots,r[n]$. 给定一个布局，每一行每一列只有一个皇后，没有冲突．但在斜线上可能产生冲突，即有 2 个或 2 个以上皇后在同一条斜线上．规定，当一条斜线上有 2 个或 2 个以上皇后时，这条斜线上的冲突数等于它的皇后数减 1；当斜线上只有一个皇后或没有皇后时，它的冲突数等于 0. 布局的冲突数等于所有斜线上的冲突数之和．一个布局是 n 后问题的解，当且仅当它的冲突数为 0.

记行号为 x，列号为 y. 在从左上到右下的斜线上 $x-y$ 为常数；在从右上到左下的斜线上 $x+y$ 为常数．这里行号从上向下、列号从左向右依次为 $1,2,\cdots,n$. 据此，容易计算给定斜线上的冲突数．

关于 n 后问题的普通局部搜索算法如下．

算法 QUEEN SEARCH：

输入：皇后数 n.

输出：无冲突布局 r.

步 1 随机生成一个初始布局 r，即 n 个数的排列 $r[1],r[2],\cdots,r[n]$.

步 2 for 每一对 $i,j(1\leq i<j\leq n)$ do

if 交换 $r[i]$ 和 $r[j]$ 使冲突数减少

then 交换 $r[i]$与 $r[j]$且 goto 2.

步 3 if 冲突数大于零 goto 1.

步 4 return r.

这个算法存在两个问题严重地影响了算法的效率. 第一个问题是,步骤 1 随机生成的初始布局的冲突数太大,给后面的搜索过程带来很大的负担;当 n 很大时,这个冲突数的期望值接近 $0.5285n$. 第二个问题是,邻域的大小为 $O(n^2)$;当 n 很大时,搜索整个邻域很费时间.

索西克与顾钧采取了两项措施:①生成冲突很少的初始布局. ②采用随机搜索. 为了描述算法,要使用几个函数和过程:

函数 Random(x,n) 返回一个 x 与 n 之间的随机整数.

过程 Swap(i,j) 交换 $r[i]$和 $r[j]$.

函数 Partial-Collisions(i) 返回经过位置$(i,r[i])$的两条斜线上从左至该位置为止的冲突数.

函数 Total-Collisions(i) 返回经过位置$(i,r[i])$的两条斜线上的冲突数.

算法 FAST RANDOMIZED QUEENS SEARCH:

输入:皇后数 n.

输出:无突冲布局 r.

步 1 Initial-Search(j,r).

步 2 Final-Search(j,r).

过程 Initial-Search(j,r)

(1-1) for $i=1$ to n do $r[i]\leftarrow i$.

(1-2) $j\leftarrow 1$.

(1-3) for $i=1$ to $3.08n$ do

(1-3-1) $m\leftarrow$Random(j,n).

(1-3-2) Swap(j,m).

(1-3-3) if Partial-Collisions$(j)=0$ then $j\leftarrow j+1$ else Swap (j,m).

(1-4) for $i=j$ to n do

  (1-4-1) $m\leftarrow$Random(i,n).

  (1-4-2) Swap(i,m).

过程 Final-Search (j,r)

(2-1) $m\leftarrow 0$.

(2-2) for $i=j$ to n do

(2-2-1) $bl\leftarrow$Total-Collisions$(i)>0$.

(2-2-2) while bl do

(2-2-2-1) $k\leftarrow$Random$(1,n)$.

(2-2-2-2) Swap(i,k).

(2-2-2-3) $m\leftarrow m+1$.

(2-2-2-4) $bl\leftarrow$(Total-Collisions$(i)>0$) or (Total-Collisions$(k)>0$).

(2-2-2-5) if bl then Swap(i,k).

(2-2-2-6) if $m \geq 7000$ goto 1.

(2-3) return r.

算法分两阶段进行．第一阶段 Initial-Search(j, r)生成一个随机的初始布局 r，这个布局在第 j 列的左边部分(不含第 j 列)没有冲突．第二阶段 Final-Search(j, r)从第 j 列到第 n 列逐列进行搜索．

过程 Initial-Search(j, r)在步骤(1-3)确定的皇后位置没有冲突．步骤(1-4)确定的皇后位置(第 j 列到第 n 列的皇后)可能有冲突．若不限制步骤(1-3)中的循环次数，直到产生一个无冲突的布局，则循环次数的期望值等于 $3.08n$．因而，以这个值作为循环次数的上限．实验结果和理论分析都表明可以把 Initial-Search 生成的初始布局的冲突数近似作为一个不大的常数．当 $n = 10^6$ 时，也只有几十个冲突．

在第二阶段，实现一个成功的交换所需的交换次数的期望值等于 72．限定交换次数的上界．若在规定的界限内没有得到问题的解，则重新从头开始．这个上界可取为 7000．实验结果表明，当 $n > 400$ 时为了得到一个解差不多只需要一个初始布局．

当 $n < 200$ 时，在第二阶段采用随机交换的效果不好，应改用检查所有可能的交换．若对于某一个 i，所有可能的交换都失败，则这个部分布局不可能扩充成一个解，计算应重新从头开始．

算法运行的时间主要花在第一阶段，与 n 是线性关系．第二阶段的运行时间差不多是一个常数，当 n 很大时是微不足道的．这是一个线性期望时间的随机局部搜索算法．

4.5 可满足性问题的局部搜索算法

可满足性问题(satisfiability problem，简记作 SAT)：给定有穷的变量集 V 和 V 上有穷的子句集 C，这里每一个变量 $x \in V$ 和它的否定 $\neg x$ 称作一个文字，若干文字的析取称作一个子句．设 V 的真假值赋值 $T: V \to \{0, 1\}$．对于文字 z，如果当 $z = x$ 时 $T(z) = 1$，当 $z = \neg x$ 时 $T(z) = 0$，其中 $x \in V$，则称 T 满足 z．T 满足子句 c 当且仅当 T 至少满足 c 中的一个文字．如果 T 满足子句集 C 中的所有子句，则称 T 满足子句集 C．如果存在满足子句集 C 的赋值，则称 C 是可满足的．问题是，判断 C 是否是可满足的，并且当 C 是可满足的时候求满足 C 的赋值 T.

子句集 C 的可满足性对应于命题逻辑中合取范式(CNF)的可满足性．由于与推理的直接联系，可满足性是命题逻辑和人工智能中的一个基本问题．这个问题是 NP 难的．

塞尔曼(B. Selman) 等人对 SAT 问题的局部搜索算法做了深入研究，提出一系列的改进策略，成功地求解了几千个变量的困难的 SAT 实例．这些策略是针对 SAT 问题提出的，同时又具有相当普遍的意义．

取 C 中不被满足的子句数作为目标函数，把问题转化成最小化问题．当目标函数值等于 0 时，赋值 T 满足 C．赋值 T 的邻域是改变 T 对一个变量的赋值所能

得到的所有赋值．

算法 GSAT：

输入：变量集 V 上的子句集 C，参数 MAX-FLIPS，MAX-TRIES．

输出：满足 C 的赋值 T(如果找到的话)或"no"．

步 1　for $i=1$ to MAX-TRIES do

(1-1) 令 $T \leftarrow$ 对 V 的随机赋值．

(1-2) for $j=1$ to MAX-FLIPS do

(1-2-1) if T 满足 C then return T．

(1-2-2) 取变量 x 使得改变 x 的值后 C 中满足的子句数增加最多．

(1-2-3) 令 $T \leftarrow$ 改变 T 关于 x 的值．

步 2　return "no"．

算法 GSAT 与普通的局部搜索算法有两点不同．第一，在步骤(1-2-2)，当有多个这样的变量时，算法随机地取其中的一个，这使得搜索过程不是完全确定的．第二，允许"平移"和"上升"．在骤步(1-2-2)，当满足的子句数增加的最大值大于零时，算法采用的是最速下降策略；当这个最大值小于等于零时，到达局部极小点．普通局部搜索算法终止这一轮试验，得到一个局部最优解，如果需要的话，再重新下一轮试验．而 GSAT 此时继续搜索，当最大值等于零时，相当于做一步"平移"；当最大值小于零时，相当于做一步"上升移动"．实验表明，上升移动是罕见的，平移的作用是跳出局部极小点，改善算法得到的解．

由于允许平移和上升移动，算法不能在局部极小点处自动终止搜索，为此，需要引入两个参数控制计算的进程．把从一个初始赋值开始到本次局部搜索终止称作一次试验．MAX-FLIPS 是在一次试验中改变变量值的最大次数，MAX-TRIES 是允许的最大试验次数．经验表明，MAX-TRIES 取变量数的不大的倍数就够了，而 MAX-FLIPS 则视使用者打算花多少时间去求一个赋值，与具体的应用有关，可以通过实验摸索．

当 C 不可满足时，GSAT 一定返回"no"．但是，当 C 可满足时，GSAT 不能保证找到满足 C 的赋值 T，尽管出现这种情况的概率非常小．GSAT 是一个不完全算法．

采用下述策略可以进一步改进算法 GSAT：

1. 加权策略

给每一个子句赋一个权，目标函数定义为不满足子句的权之和．开始时，子句的权初始化为 1．当到达局部极小点时，不满足子句的权加 1．接下去可以有两种做法：第一种是重新取一个随机初始赋值开始下一轮局部搜索，第二种是继续接着往下搜索．加权的作用是解决子句之间的不对称性．局部极小点相当于一个凹坑，通过加权填平凹坑．

2. 随机游动策略

事先给定概率 $p(0 \leqslant p \leqslant 1)$．在每一步以概率 p 执行随机游动，即从任一不满足的子句中取一个变量，改变它的赋值．以概率 $1-p$ 执行标准的 GSAT 运算，即取变量 x，使得 x 的赋值改变后满足的子句数增加最多，改变 x 的赋值．实验表明，对于随机 3-SAT 实例(每个子句恰好含有 3 个文字)，取 $p=0.5$—0.6 最好．

3. 大步跳转策略

这个策略也是用来跳出局部极小点．做法如下：当到达局部极小点后，从任一不满足的子句中取一个变量，改变这个变量的值．这样可能使一些原先已满足的子句变成不满足的．从这样的子句中任取一个另外的变量（不是已经取过的变量），改变它的值．重复这个过程，直到没有符合条件的变量为止．然后以这个赋值作为初始赋值重新开始局部搜索．

参考文献

1 M.R.加里，D.S.约翰逊著．张立昂，沈泓，毕源章译，吴允曾校．计算机和难解性：NP完全性理论导引．北京：科学出版社，1987.

2 C.H.Papadimitriou，K.Steiglitz著．刘振宏，蔡茂诚译．组合最优化：算法和复杂性．北京：清华大学出版社，1988.

3 Christos H．Papadimitriou．Computational complexity．Reading MA：Addison - Wesley，1994.

4 Rok Sosic，Jun Gu．Efficient local search with conflict minimization：a case study of the n-queens problem．IEEE Transactions on Knowledge and Data Engineering，1994(6)：661～668.

5 Bart Selman，H．Levesque，and D．Mitchell．A new method for solving hard satisfiability problems．Proceedings AAAI-92，1992：440～446.

·计算机数学卷·

第14篇

遗传算法

编　者　刘宝碇

审校者　卢开澄

目　录

引　言

遗传算法(genetic algorithm)是一种模拟自然进化过程的随机搜索方法.根据达尔文的生物进化学说,地球上的生物繁殖后代,既有遗传也有变异,依"优胜劣汰"的原则使生物种群不断进化.为寻求最优化问题的解,遗传算法仿效这一过程,模拟生物的遗传与变异,采取适者生存的原则而搜索到问题的最优解.

1975年Holland出版了专著Adaptation in Natural and Artificial Systems,从而遗传算法得到了系统地总结,这也标志着遗传算法得到了正式承认.此后的二十几年中,遗传算法得到了迅猛的发展.在解决全局优化问题方面,遗传算法显示了非常广泛的应用前景,并已应用到最优控制、运输问题、流动推销员问题、时间表问题、排序问题、生产计划、资源分配、统计及模式识别等诸多领域.

在优化问题中,如果目标函数是多峰的,或者搜索空间不规则,就要求所使用的算法必须具有高度的鲁棒性,以避免在局部最优解附近徘徊.遗传算法的优点恰好是擅长全局搜索.另外,遗传算法本身并不要求对优化问题的性质作一些深入的数学分析,从而对那些不太熟悉数学理论和算法的使用者来说,无疑是方便的.

1　遗传算法

生物遗传物质的主要载体是**染色体**.在遗传算法中,染色体通常是一串数据(或数组),用来作为优化问题解的代码,其本身不一定是解.遗传算法一般经过这样几个过程:首先,随机产生一定数目的初始染色体,这些随机产生的染色体组成一个**种群**.种群中染色体的数目称为种群的大小或种群规模.然后,用评价函数来评价每一个染色体的优劣,也就是计算染色体对环境的适应程度(称为**适应度**),用来作为以后进行遗传操作的依据.接着,执行选择过程,其目的是为了从当前种群中选出优良的染色体,使它们成为新一代的种群.判断染色体优良与否的准则就是各自的适应度,即染色体的适应度越高,其被选择的机会就越大.通过选择过程,产生一个新的种群,再对这个新的种群进行交叉操作,它是遗传算法中主要的遗传操作之一.接着进行变异操作,变异操作的目的是挖掘种群中个体的多样性,克服有可能限于局部最优解的弊病.经过上述运算产生的染色体称为后代.然后,对新的种群(即后代)重复进行选择、交叉和变异操作,经过给定代数的进化以后,把最好的染色体作为优化问题的最优解.

有时遗传算法也称为进化规划、进化策略或遗传规划.虽然这些概念的内涵有一定的差别,但本质上都是基于进化思想的,因此又统称为**进化算法**.感兴趣的读者可进一步阅读本篇后所列的参考文献.

在阅读本篇介绍的遗传算法之前,提醒读者注意,本篇是针对如下的单目标规

划、多目标规划以及目标规划模型介绍遗传算法的. 这里, 单目标数学规划的一般形式表示为

$$\begin{cases} \max f(\boldsymbol{x}), \\ \text{s.t. } g_j(\boldsymbol{x}) \leqslant 0, \quad j = 1,2,\cdots,p, \end{cases} \tag{1-1}$$

即在一组约束条件 $g_j(\boldsymbol{x}) \leqslant 0, j = 1,2,\cdots,p$ 下, 使目标函数 $f(\boldsymbol{x})$ 达到极大(有时也可能是极小). 其中 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ 是一个 n 维决策向量, $f(\boldsymbol{x})$ 和 $g_j(\boldsymbol{x})(j = 1, 2, \cdots, p)$ 是普通的实值函数. 在同样的一组约束条件下, 使 m 个不同的目标函数达到最优的问题可用多目标规划描述, 其一般形式为

$$\begin{cases} \max[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})], \\ \text{s.t. } g_j(\boldsymbol{x}) \leqslant 0, j = 1,2,\cdots,p, \end{cases} \tag{1-2}$$

其中 $f_i(\boldsymbol{x})(i = 1,2,\cdots,m)$ 和 $g_j(\boldsymbol{x})(j = 1,2,\cdots,p)$ 是决策向量 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ 的普通实值函数. 根据决策者给定的目标值和优先结构, 非线性目标规划的一般形式可以表示为

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-), \\ \text{s.t. } f_i(\boldsymbol{x}) + d_i^- - d_i^+ = b_i, & i = 1,2,\cdots,m, \\ \qquad g_j(\boldsymbol{x}) \leqslant 0, & j = 1,2,\cdots,p, \\ \qquad d_i^-, d_i^+ \geqslant 0, & i = 1,2,\cdots,m, \end{cases} \tag{1-3}$$

其中 P_j 为优先因子, 表示各个目标的相对重要性, 且对所有的 j, 有 $P_j \gg P_{j+1}$, u_{ij} 是对应优先因子 j 的第 i 个目标正偏差的权重因子, v_{ij} 是对应优先因子 j 的第 i 个目标负偏差的权重因子, d_i^+ 是目标 i 偏离目标值的正偏差, d_i^- 是目标 i 偏离目标值的负偏差, $\boldsymbol{x}$ 是 n 维决策向量, f_i 是目标约束中的函数, g_j 是系统约束中的函数, b_i 是目标 i 的目标值, l 是优先级个数, m 是目标约束个数, p 是系统约束个数.

在上面提到的三种数学规划模型中, 目标函数不一定是单峰的, 可行集也不一定是凸的. 另外, 值得一提的是, 在上述模型中, 认为不存在等式约束, 正如在后面讨论中提到的那样, 等式约束一般是可以消除的.

1.1 表达方式

在遗传算法中, 一个关键问题是如何将一个解表示成染色体形式, 即如何编码. 早期的研究工作是将一个问题的解表示成二进制向量形式. 但对于很多最优化问题, 二进制向量表达方式经常难以应用. 于是很多新的编码也就应运而生了. 除了二进制向量表达方式之外, 浮点向量也是常用的表达方式.

一方面有原始的解空间, 另一方面要设计与之对应的染色体(编码)空间. 解空间与染色体空间的映射应尽量满足一一对应关系. 表达方式的好坏直接影响到进化过程的快慢. 一个好的编码方式能够更快地找到最优解. 然而, 并没有一个编码的一般性指导原则, 编码技术与其说是科学, 不如说是技巧.

使用二进制向量作为一个染色体来表示解的真实值, 则向量的长度依赖于要

求的精度.例如,假设在区间[1,3]上优化(极大或极小)一个一元实值函数 $f(x)$,并要求最优解的精度为小数点后六位.显然,变量 x 的可行集是长度为 2 的区间,于是,为使精度达到小数点后六位,该区间[1,3]至少应分成2 000 000等份.也就是说,若区间[1,3]由这些等分点代替,则在这些离散点上找到的最优解的精度可达到小数点后六位.由于

$$1\ 048\ 576 = 2^{20} < 2\ 000\ 000 < 2^{21} = 2\ 097\ 152,$$

我们必须采用 21 位的二进制向量去表示上述问题的解,这也意味着将把区间等分成 2 097 151 份.记这样的二进制向量为 $\langle b_{20}b_{19}\cdots b_1b_0\rangle$,所有这样的二进制向量构成一个编码空间.该编码空间可按如下方式映射到区间[1,3]上:

步 1　将二进制向量 $\langle b_{20}b_{19}\cdots b_1b_0\rangle$ 转化为十进制数,

$$(\langle b_{20}b_{19}\cdots b_1b_0\rangle)_2 = \left(\sum_{k=0}^{20} b_k \cdot 2^k\right)_{10} = x'.$$

步 2　由于 x' 是介于 0 与 $\sum_{k=0}^{20} 2^k = 2^{21} - 1$ 之间的数,可将 x' 按如下方式映射到区间[1,3]上得到 x,

$$x = 1 + (3-1) \cdot \frac{x'}{2^{21}-1}.$$

其中 1 为左边界,3 为右边界.

这意味着,若希望在区间[1,3]上找到精确到小数点后六位的最优解,则可在 21 位的二进制向量中找出一个来表示这个最优解.

但使用二进制代码的必要性已经受到了一些批评.在求解复杂优化问题时,二进制向量表达方式有时不太方便,另一种表示方法是用浮点向量,每一个染色体由一个浮点向量表示,其长度与解向量相同.若向量 $x = (x_1, x_2, \cdots, x_n)$ 是最优化问题的解,其中 n 是维数,则相应的染色体也表示为 $V = (x_1, x_2, \cdots, x_n)$.

1.2　处理约束条件

在遗传算法中,如果设计的初始化过程产生了非可行的染色体,或者以后的遗传操作产生了非可行的后代,应该如何处理这种情况呢?

1. 删除等式约束

在数学规划模型中,若等式约束存在,一般来说,任何的遗传操作都不可能产生可行的后代.当存在一些等式约束,例如,$h_k(x) = 0, k = 1,2,\cdots,q$,可以通过解这些约束构成的方程组,用其它的变量替换其中的 q 个变量,以消除 q 个等式约束.例如,不失一般性,其前 q 个变量通过解方程组得到 $x_k = \hat{h}_k(x_{q+1}, x_{q+2}, \cdots, x_n)$, $k = 1,2,\cdots,q$,则可以将最优化模型中的变量 $x_1, x_2, \cdots, x_q$ 由其它变量表示,于是原问题化成一个$(n-q)$维的只有不等式约束的新问题.

2. 拒绝策略

当产生非可行解时,拒绝策略舍弃所有的非可行解.当可行域为凸集时,这种

方式通常是有效的.然而,当可行集非凸时,也许由非可行解到达最优解更"容易",所以拒绝策略也未必总是好策略.

3. 修复策略

当产生非可行解时,可以通过某种方式将非可行解化成一个可行解.对于一些组合优化问题,此种策略是有效的.然而它的缺点是算法与问题有关,且由非可行解到可行解的转化过程也未必容易设计.

4. 改善遗传算子策略

有时可以对给定的问题采用一些特殊的表达方式和特别设计的算子以保证后代的可行性.

5. 惩罚策略

对于某些约束问题,遗传算子会产生大量的非可行解.当约束集足够复杂时,也许上述策略不能保证后代的可行性.于是可以采取惩罚策略,这非常类似于非线性规划技术中的惩罚函数法.惩罚策略的关键问题是设计一个惩罚函数 $p(x)$,它将迫使遗传搜索由非可行解走向最优解.然而,并没有一个一般性的设计惩罚函数的指导原则.设计惩罚函数依赖于原问题的性质,例如,对于最优化问题(1-1)可以定义惩罚函数为

$$p(x) = \sum_{j=1}^{p} \lambda_j \bar{g}_j(x),$$

其中 $\lambda_1, \lambda_2, \cdots, \lambda_p$ 是 p 个常数,而

$$\bar{g}_j(x) = \begin{cases} 0, & g_j(x) \leqslant 0, \\ -g_j(x), & g_j(x) > 0 \end{cases} \quad (j = 1, 2, \cdots, p).$$

于是原约束优化问题变成一个无约束优化问题,即 $\max[f(x) + p(x)]$.

最后值得注意的是,在设计程序时,应当注意到一些隐含约束,即有些点虽然是可行解,但不可能是最优解.例如,数学规划模型

$$\max_{x \in \mathbf{R}} \exp(-x^2)$$

的可行集是全体实数 $\mathbf{R}$.但是,通过数学分析可以确认,最优解一定在区间 $[-5, +5]$ 上.所以,为减少程序的搜索空间,应当增加约束

$$-5 \leqslant x \leqslant 5.$$

这一类隐含约束一般不难发现,尤其对实际的管理决策问题更是如此.因此,尽可能地增加隐含条件以减少搜索空间,可以大大加快求解问题的进化过程.

1.3　初始化过程

定义整数 N 作为种群规模,即染色体的个数,并且随机产生 N 个初始染色体.一般情况下,由于优化问题的复杂性,解析地产生可行的染色体是困难的.此时,可以采用下述两种方法之一作为初始化过程.具体实施时依赖于事先能够提供的信息.

第一种方法　设决策者能够给出可行集中的一个内点,记为 V_0.定义一个足

够大的数 M,以保证遗传操作遍及整个可行集.此大数 M 不仅在初始化过程中使用而且也在变异操作中使用.注意此大数 M 的选择是与优化问题有关的.按照下面的方法产生 N 个染色体:在 $\mathbf{R}^n$ 中,随机选择一个方向 d,如果 V_0+Md 满足不等式约束,则将 $V=V_0+Md$ 作为一个染色体,否则,置 M 为0和 M 之间的一个随机数,直到 V_0+Md 可行为止.由于 V_0 是内点,所以在有限步内可以找到满足不等式约束的可行解.重复以上过程 N 次,从而产生 N 个初始染色体 $V_1,V_2,\cdots,V_N$.

第二种方法 如果决策者不能给出这样的内点,但可以确定一个包含最优解(不一定是整个可行集)的区域.显然,在任何情况下,只要不介意所给区域过大,决策者总能提供这样的区域,一般情况下,把该区域设计成一个易于抽样的形状,如可以设计成一个 n 维超立方体.从这个超立方体中产生一个随机点,并检验其可行性.如果可行,则作为一个染色体,否则,从超立方体中重新产生随机点,直到得到可行解为止.顺便说一句,只要所产生的是随机整数,这种方法同样也适合组合优化问题.重复以上过程 N 次,则得到 N 个初始可行的染色体 $V_1,V_2,\cdots,V_N$.

1.4 评价函数

评价函数(用 eval(V) 表示)用来对种群中的每个染色体 V 设定一个概率,以使该染色体被选择的可能性与其种群中其它染色体的适应性成比例.染色体的适应性越强,被选择的可能性也应越大.

第一种方法是设目前该代中的染色体为 $V_1,V_2,\cdots,V_N$,根据染色体的序进行再生分配,而不是根据其实际的目标值.无论何种数学规划(单目标、多目标或目标规划)都可以作一合理假设,即在染色体 $V_1,V_2,\cdots,V_N$ 中,决策者可以给出一个序的关系,使染色体由好到坏进行重排,也就是说,一个染色体越好,其序号越小.例如,对于单目标极大化问题,染色体所对应的目标值越高,说明该染色体越好;对于多目标规划问题,也可以定义一个偏好函数去评价染色体;对于目标规划,将采用如下的序关系重排染色体;如果两个染色体在高优先级上目标值相等,则在当前优先级上目标值小的染色体为优;如果两个染色体在所有优先级上目标值都相同,则这两个染色体是一样的,于是可以随机重排它们.后面将通过一些例子来说明这一点.设参数 $a\in(0,1)$ 给定,定义基于序的评价函数为

$$\mathrm{eval}(V_i)=a(1-a)^{i-1},\quad i=1,2,\cdots,N.$$

这里 $i=1$ 意味着该染色体是最好的,$i=N$ 说明是最差的.

第二种方法是通过对适应度的适当缩放调整(称为适应度定标)来设计评价函数.用 $f_1,f_2,\cdots,f_N$(即染色体 $V_1,V_2,\cdots,V_N$ 各自的目标值)表示原来的适应度.Goldberg[3] 提出了一种线性适应度定标方案,

$$f'_i=af_i+b,\quad i=1,2,\cdots,N,$$

其中 f'_i,为新的适应度,a 和 b 为参数.这种方法实际上假定使用者了解目标函数的性质,从而才能设计合理的参数 a 和 b.在这种情况下,评价函数定义为

$$\mathrm{eval}(V_i)=f'_i/\sum_{j=1}^{N}f'_j,\quad i=1,2,\cdots,N.$$

第三种方法是基于指数适应度的评价函数,它介于基于序的评价函数和线性适应度定标方案之间.首先,定义三个优先参数 p_1、p_0 和 p_2($0 < p_1 < p_0 < p_2 < 1$),以确定三个临界数 u_1、u_0 和 u_2(取自于 N 个染色体目标值构成的集合),使在目前的 N 个染色体构成的集合中分别有($p_1 \cdot N$),($p_0 \cdot N$) 和($p_2 \cdot N$) 个染色体的目标值分别小于 u_1、u_0 和 u_2.

对极大化问题,原来的适应度 u_1 映射到 $e^{-1} \approx 0.37$,u_0 映射到 1,u_2 映射到 $2 - e^{-1} \approx 1.63$,则原来的适应度 u 和指数适应度 u' 之间的关系为

$$u' = \begin{cases} \exp\left(-\dfrac{u - u_0}{u_1 - u_0}\right), & u < u_0 \\ 2 - \exp\left(-\dfrac{u - u_0}{u_2 - u_0}\right), & u \geq u_0 \end{cases}$$

此时,定义如下的评价函数,

$$\text{eval}(V_i) = u'_i \Big/ \sum_{j=1}^{N} u'_j, \quad i = 1,2,\cdots,N,$$

其中 u'_i 分别是染色体 V_i 各自的指数适应度.

除此之外,还有许多类型的评价函数,有兴趣的读者可参阅有关的参考文献.

1.5　选择过程

选择过程是以旋转赌轮 N 次为基础的.每次旋转都为新的种群选择一个染色体.赌轮按每个染色体的适应度来选择染色体.无论使用哪一种评价函数,选择过程总可以陈述如下.

步 1　对每个染色体 V_i,计算累积概率 q_i,$i = 0,1,2,\cdots,N$,

$$\begin{cases} q_0 = 0, \\ q_i = \sum_{j=1}^{i} \text{eval}(V_j), \quad i = 1,2,\cdots,N. \end{cases}$$

步 2　从区间 $(0, q_N]$ 中产生一个随机数 r.

步 3　若 $q_{i-1} < r \leq q_i$,则选择第 i 个染色体 V_i($1 \leq i \leq N$).

步 4　重复步 2 和步 3 共 N 次,这样可以得到 N 个复制的染色体.

在上述过程中,并没有要求满足条件 $q_N = 1$.事实上,可以将所有的 q_i,$i = 1,2,\cdots,N$ 除以 q_N 使得 $q_N = 1$.新得到的概率同样与适应度成比例.只要不介意概率方面解释上的困难,这一点并没有在进化过程中产生任何影响.

1.6　交叉操作

首先定义参数 P_c 作为交叉操作的概率,这个概率说明种群中有期望值为 P_cN 个染色体将进行交叉操作.

为确定交叉操作的父代,从 $i = 1$ 到 N 重复以下过程:从[0,1]中产生随机数

r,如果 $r < P_c$,则选择 V_i 作为一个父代.

用 $V'_1, V'_2, \cdots$ 表示上面选择的父代,并把它们随机分成下面的对

$$(V'_1, V'_2), \quad (V'_3, V'_4), \quad (V'_5, V'_6), \cdots,$$

当父代个数为奇数时,既可以去掉一个染色体也可以再选择一个染色体以保证两两成对.下面以(V'_1, V'_2)为例解释怎样对上面所有的对进行交叉操作.

第一种方式是简单交叉.设染色体 $V'_1 = (a_1, a_2, \cdots, a_n)$, $V'_2 = (b_1, b_2, \cdots, b_n)$.随机产生两个介于 1 与 n 之间的交叉点 j 和 k($j \leqslant k$),通过交换染色体 V'_1 和 V'_2 的第 j 至第 k 个基因来形成两个后代,即

$$X = (a_1, \cdots, a_{j-1}, b_j, \cdots, b_k, a_{k+1}, \cdots, a_n),$$
$$Y = (b_1, \cdots, b_{j-1}, a_j, \cdots, a_k, b_{k+1}, \cdots, b_n).$$

第二种方式称为算术交叉.首先,从开区间(0,1)中产生一个随机数 c,然后,按下列形式在 V'_1 和 V'_2 之间进行交叉操作,并产生如下的两个后代 X 和 Y,

$$X = c \cdot V'_1 + (1 - c) \cdot V'_2, \quad Y = (1 - c) \cdot V'_1 + c \cdot V'_2.$$

如果可行集是凸的,这种凸组合交叉运算在两个父代可行的情况下,能够保证两个后代也是可行的.但是,在许多情况下,可行集不一定是凸的,或很难验证其凸性,此时必须检验每一后代的可行性.如果两个后代均可行,则用它们代替其父代,否则,保留其中可行的(如果存在的话),然后,产生新的随机数 c,重新进行交叉操作,直到得到两个可行的后代或循环给定次数为止.无论如何,仅用可行的后代取代其父代.

当新产生的后代不可行时,也可以采取一些修复策略使之变成可行染色体.

1.7 变异操作

定义参数 P_m 作为遗传系统中的变异概率,这个概率表明,种群中将有期望值为 $P_m N$ 个染色体用来进行变异操作.

类似于交叉操作中选择父代的过程,由 $i = 1$ 到 N,重复下列过程:从区间[0,1]中产生随机数 r,如果 $r < P_m$,则选择染色体 V_i 作为变异的父代.对每一个选择的父代,用 $V = (x_1, x_2, \cdots, x_n)$ 表示,按下列方法进行变异.

第一种方式是首先选择一个介于 1 与 n 之间的变异点 k,然后在区间$[x_k^{\min}, x_k^{\max}]$上随机产生一个随机数 x'_k,其中 $x_k^{\max}$ 和 $x_k^{\min}$ 是第 k 个基因的上、下界.将 x'_k 代替染色体 V 中的第 k 个基因 x_k,从而形成一个后代 $V' = (x_1, \cdots, x_{k-1}, x'_k, x_{k+1}, \cdots, x_n)$.

第二种方式是在 $\mathbf{R}^n$ 中随机选择变异方向 d,如果 $V + Md$ 是不可行的,那么,置 M 为 0 和 M 之间的随机数,直到其可行为止.其中 M 是在初始化过程中定义的一个足够大的数.如果在预先给定的迭代次数之内没有找到可行解,则置 $M = 0$.无论 M 为何值,总用

$$X = V + Md$$

代替 V.

1.8 遗传算法程序

经过选择、交叉和变异操作，得到一个新的种群，准备进行下一代进化. 对上述步骤经过给定的循环次数之后，遗传算法终止. 对一般优化问题，遗传算法可以归纳如下：

遗传算法程序：

输入参数 N，P_c，P_m；

通过初始化过程产生 N 个染色体.

重复：

对染色体进行交叉和变异操作；

计算所有染色体的评价函数；

根据某种抽样机制选择染色体；

直到满足终止条件.

大家知道，最好的染色体不一定出现在最后一代中，所以在进化开始，必须把最好的染色体保留下来，记为 V_0，如果在新的种群中又发现了更好的染色体，则用它代替原来的染色体 V_0. 在进化完成之后，这个染色体就可以看做是优化问题的解.

2 遗传算法与上升法的比较

上升法是非线性规划求解技术：直接法、梯度法和黑塞（Hessian）法的通称. 上升法首先在最优解可能存在的地方选择一个初始点，然后通过分析目标函数的特性，由初始点移到一个新的点，然后再继续这个过程直至找到最优解. 为了更好地理解遗传算法，现在把上升法和遗传算法作一比较.

上升法的搜索过程是确定的，通过产生一系列的点收敛到最优解（有时是局部最优解），而遗传算法的搜索过程是随机的，它产生一系列随机种群序列. 二者的主要差异可以归纳如下两点：

(1) 上升法的初始点仅有一个，由决策者给出；遗传算法的初始点则有多个（即初始种群），随机产生.

(2) 通过分析目标函数的特性，上升法由上一点产生一个新的点；遗传算法通过遗传操作，在当前的种群中经过交叉、变异和选择产生下一代种群.

对同一优化问题，遗传算法所使用的机时比上升法所花费的机时要多得多. 但遗传算法可以处理一些上升法不能解决的复杂的优化问题. 另外，遗传算法目前还不能处理大规模的问题.

我们已将遗传算法编写成 C 语言程序. 在后面给出的数值例子中，所使用的参数如下：种群规模为30，交叉概率为0.2，变异概率为0.5，而评价函数中的参数 a 为0.05. 遗传算法常常因为这些参数的设置受到各种各样的批评. 遗传算法对于这些

参数的设置是非常鲁棒的,改变这些参数对所得的结果不会有太大的影响.

例 1 单目标规划.考虑非凸集合上的优化问题

$$\begin{cases} \max f(x) = \dfrac{x_1^2 x_2 x_3^2}{2x_1^3 x_3^2 + 3x_1^2 x_2^2 + 2x_2^2 x_3^3 + x_1^3 x_2^2 x_3^2}, \\ \text{s.t.}\ x_1^2 + x_2^2 + x_3^2 \geqslant 1, \\ \quad x_1^2 + x_2^2 + x_3^2 \leqslant 4, \\ \quad x_1, x_2, x_3 > 0. \end{cases}$$

已经知道目标函数的最大值为$f(x) = 0.1537$,图 2-1 中的阴影部分表示可行集在$x_3 = 0$处的横截面.从图中可以看到,可行集显然是非凸的.

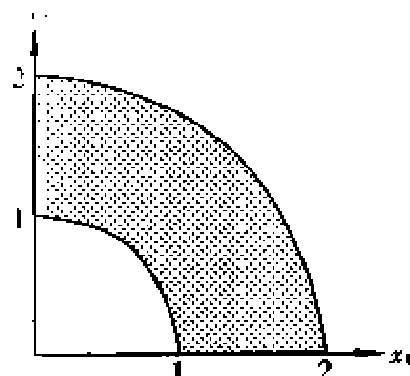


图 2-1

下面用遗传算法求解此问题.用染色体$V = (x_1, x_2, x_3)$来作为解的代码.染色体$V = (x_1, x_2, x_3)$的可行性由下面的检验函数检验:

如果($x_1 \leqslant 0 \parallel x_2 \leqslant 0 \parallel x_3 \leqslant 0$),返回 0;

如果($x_1^2 + x_2^2 + x_3^2 < 1$),返回 0;

如果($x_1^2 + x_2^2 + x_3^2 > 4$),返回 0;

返回 1.

其中检验函数值 0 表示不可行,1 表示可行.容易知道可行集包含于下列超几何体中

$$\Omega = \{(x_1, x_2, x_3) \mid 0 \leqslant x_1 \leqslant 2, 0 \leqslant x_2 \leqslant 2, 0 \leqslant x_3 \leqslant 2\}.$$

可以很容易地从这样的超几何体中抽样,例如,取

$$x_1 = u(0,2), \quad x_2 = u(0,2), \quad x_3 = u(0,2),$$

其中函数$u(a,b)$用来产生区间$[a,b]$上的均匀分布的随机数.如果这个染色体不可行,则拒绝接受,重新产生一个新的染色体;如果产生的染色体可行,则接受它作为种群的一名成员.经过有限次抽样以后,得到 30 个可行的染色体:

$$V_1 = (0.3903, 0.6723, 1.2507),\quad V_2 = (0.9167, 0.2930, 0.3297),$$
$$V_3 = (0.2373, 0.1267, 1.7370),\quad V_4 = (0.8523, 0.9683, 1.4477),$$
$$V_5 = (0.1280, 0.8337, 1.1807),\quad V_6 = (0.3283, 0.6830, 1.8263),$$
$$V_7 = (1.1223, 0.6363, 1.2303),\quad V_8 = (0.5020, 0.8447, 1.0840),$$
$$V_9 = (0.0490, 1.7077, 0.2813),\quad V_{10} = (0.5643, 0.5450, 1.6913),$$
$$V_{11} = (1.1430, 0.6000, 0.3623),\quad V_{12} = (1.6243, 1.0153, 0.5573),$$
$$V_{13} = (0.7953, 1.3563, 1.1223),\quad V_{14} = (0.1240, 1.7903, 0.5593),$$
$$V_{15} = (1.2320, 0.0733, 0.9930),\quad V_{16} = (1.4473, 1.3397, 0.2947),$$
$$V_{17} = (0.3960, 0.6173, 1.2623),\quad V_{18} = (0.5420, 0.4000, 1.6593),$$
$$V_{19} = (0.1517, 1.0047, 0.5590),\quad V_{20} = (1.2550, 1.2957, 0.6413),$$
$$V_{21} = (0.1313, 0.8217, 1.4523),\quad V_{22} = (0.2383, 1.2930, 0.3637),$$
$$V_{23} = (1.3047, 0.4163, 0.4673),\quad V_{24} = (1.7893, 0.5220, 0.4343),$$

$V_{25}=(1.1910,0.1460,0.5890)$, $V_{26}=(0.6023,1.3187,0.3897)$,

$V_{27}=(0.9907,0.8447,0.9030)$, $V_{28}=(0.7467,1.2017,1.0873)$,

$V_{29}=(0.9263,1.5153,0.0503)$, $V_{30}=(0.0823,0.1867,1.1217)$.

直接计算,得到如下的初始适应度的值,即目标值:

$f(V_1)=0.0727$, $f(V_2)=0.0674$, $f(V_3)=0.0854$,

$f(V_4)=0.1277$, $f(V_5)=0.0082$, $f(V_6)=0.0401$,

$f(V_7)=0.1482$, $f(V_8)=0.0906$, $f(V_9)=0.0022$,

$f(V_{10})=0.1144$, $f(V_{11})=0.0539$, $f(V_{12})=0.0663$,

$f(V_{13})=0.0971$, $f(V_{14})=0.0068$, $f(V_{15})=0.0294$,

$f(V_{16})=0.0197$, $f(V_{17})=0.0792$, $f(V_{18})=0.1269$,

$f(V_{19})=0.0170$, $f(V_{20})=0.0711$, $f(V_{21})=0.0071$,

$f(V_{22})=0.0215$, $f(V_{23})=0.0784$, $f(V_{24})=0.0616$,

$f(V_{25})=0.0560$, $f(V_{26})=0.0327$, $f(V_{27})=0.1275$,

$f(V_{28})=0.1013$, $f(V_{29})=0.0006$, $f(V_{30})=0.0158$.

从中可以发现,染色体 V_7 是其中最好的染色体,而染色体 V_{29} 是其中最差的.在此次进化中,保留染色体 V_7,记为 V_0.如果在以后的进化过程中发现比 V_0 更好的染色体,则用它取代 V_0.根据染色体的目标值,由好到坏重排染色体如下:

$V'_1=(1.1223,0.6363,1.2303)(V_7)$, $V'_2=(0.8523,0.9683,1.4477)(V_4)$,

$V'_3=(0.9907,0.8447,0.9030)(V_{27})$, $V'_4=(0.5420,0.4000,1.6593)(V_{18})$,

$V'_5=(0.5643,0.5450,1.6913)(V_{10})$, $V'_6=(0.7467,1.2017,1.0873)(V_{28})$,

$V'_7=(0.7953,1.3563,1.1223)(V_{13})$, $V'_8=(0.5020,0.8447,1.0840)(V_8)$,

$V'_9=(0.2373,0.1267,1.7370)(V_3)$, $V'_{10}=(0.3960,0.6173,1.2623)(V_{17})$,

$V'_{11}=(1.3047,0.4163,0.4673)(V_{23})$, $V'_{12}=(0.3903,0.6723,1.2507)(V_1)$,

$V'_{13}=(1.2550,1.2957,0.6413)(V_{20})$, $V'_{14}=(0.9167,0.2930,0.3297)(V_2)$,

$V'_{15}=(1.6243,1.0153,0.5573)(V_{12})$, $V'_{16}=(1.7893,0.5220,0.4343)(V_{24})$,

$V'_{17}=(1.1910,0.1460,0.5890)(V_{25})$, $V'_{18}=(1.1430,0.6000,0.3623)(V_{11})$,

$V'_{19}=(0.3283,0.6830,1.8263)(V_6)$, $V'_{20}=(0.6023,1.3187,0.3897)(V_{26})$,

$V'_{21}=(1.2320,0.0733,0.9930)(V_{15})$, $V'_{23}=(1.4473,1.3397,0.2947)(V_{16})$,

$V'_{22}=(0.2383,1.2930,0.3637)(V_{22})$, $V'_{24}=(0.1517,1.0047,0.5590)(V_{19})$,

$V'_{25}=(0.0823,0.1867,1.1217)(V_{30})$, $V'_{27}=(0.1313,0.8217,1.4523)(V_{21})$,

$V'_{26}=(0.1280,0.8337,1.1807)(V_5)$, $V'_{28}=(0.1240,1.7903,0.5593)(V_{14})$,

$V'_{29}=(0.0490,1.7077,0.2813)(V_9)$, $V'_{30}=(0.9263,1.5153,0.0503)(V_{29})$.

根据基于序的评价函数及 $a=0.05$,有

$$\mathrm{eval}(V'_i)=0.05\times(1-0.05)^{i-1},\quad i=1,2,\cdots,N.$$

于是得到

$q_1=0.0500$, $q_2=0.0975$, $q_3=0.1426$, $q_4=0.1855$, $q_5=0.2262$,

$q_6 = 0.2649$, $q_7 = 0.3017$, $q_8 = 0.3366$, $q_9 = 0.3698$, $q_{10} = 0.4013$,
$q_{11} = 0.4312$, $q_{12} = 0.4596$, $q_{13} = 0.4867$, $q_{14} = 0.5123$, $q_{15} = 0.5367$,
$q_{16} = 0.5599$, $q_{17} = 0.5819$, $q_{18} = 0.6028$, $q_{19} = 0.6226$, $q_{20} = 0.6415$,
$q_{21} = 0.6594$, $q_{22} = 0.6765$, $q_{23} = 0.6926$, $q_{24} = 0.7080$, $q_{25} = 0.7226$,
$q_{26} = 0.7365$, $q_{27} = 0.7497$, $q_{28} = 0.7622$, $q_{29} = 0.7741$, $q_{30} = 0.7854$.

现在准备旋转赌轮 30 次.首先由计算机在区间$(0, q_{30}] = (0, 0.7854]$上产生随机数,得到0.0328,其大于$q_0 = 0$,而小于$q_1 = 0.0500$,所以选择染色体$V'_1(V_7)$作为新种群的一名成员.第二次产生的随机数为0.1284,大于$q_2 = 0.0975$,而小于$q_3 = 0.1426$,所以$V'_3(V_{27})$也被选中.经过30次选择之后,得到一个新的种群

$V''_1 = (1.1223, 0.6363, 1.2303)$,　$V''_2 = (0.9907, 0.8447, 0.9030)$,
$V''_3 = (0.7467, 1.2017, 1.0873)$,　$V''_4 = (1.1223, 0.6363, 1.2303)$,
$V''_5 = (0.9167, 0.2930, 0.3297)$,　$V''_6 = (0.5420, 0.4000, 1.6593)$,
$V''_7 = (0.9167, 0.2930, 0.3297)$,　$V''_8 = (0.8523, 0.9683, 1.4477)$,
$V''_9 = (1.3047, 0.4163, 0.4673)$,　$V''_{10} = (0.9263, 1.5153, 0.0503)$,
$V''_{11} = (0.0823, 0.1867, 1.1217)$,　$V''_{12} = (0.5020, 0.8447, 1.0840)$,
$V''_{13} = (0.9263, 1.5153, 0.0503)$,　$V''_{14} = (0.9263, 1.5153, 0.0503)$,
$V''_{15} = (0.3960, 0.6173, 1.2623)$,　$V''_{16} = (1.7893, 0.5220, 0.4343)$,
$V''_{17} = (0.8523, 0.9683, 1.4477)$,　$V''_{18} = (0.9263, 1.5153, 0.0503)$,
$V''_{19} = (0.6023, 1.3187, 0.3897)$,　$V''_{20} = (0.1313, 0.8217, 1.4523)$,
$V''_{21} = (0.9263, 1.5153, 0.0503)$,　$V''_{22} = (1.3047, 0.4163, 0.4673)$,
$V''_{23} = (1.1223, 0.6363, 1.2303)$,　$V''_{24} = (1.6243, 1.0153, 0.5573)$,
$V''_{25} = (1.1910, 0.1460, 0.5890)$,　$V''_{26} = (0.9907, 0.8447, 0.9030)$,
$V''_{27} = (0.1517, 1.0047, 0.5590)$,　$V''_{28} = (1.1223, 0.6363, 1.2303)$,
$V''_{29} = (0.0490, 1.7077, 0.2813)$,　$V''_{30} = (1.6243, 1.0153, 0.5573)$.

接着,对新的种群进行遗传操作,即交叉和变异操作.交叉概率$P_c = 20\%$,说明平均有6个染色体进行交叉操作.从区间[0,1]上产生随机数,得0.6437,大于$P_c = 0.20$,第一个染色体V''_1没有选中.第二次产生的随机数为0.1256,小于$P_c = 0.20$,第二个染色体V''_2被选中用来作为交叉操作的一个父代.这样,经过30次,选中10个染色体

$$V''_2, V''_3, V''_4, V''_6, V''_{12}, V''_{13}, V''_{15}, V''_{16}, V''_{19}, V''_{28},$$

将它们随机分成五组

$$(V''_{16}, V''_{19}), (V''_3, V''_6), (V''_{15}, V''_2), (V''_4, V''_{12}), (V''_{13}, V''_{28}).$$

由于被选中的染色体是偶数个,所以容易配对,若是奇数个,只需去掉一个即可.通过交叉操作,得到如下的一个种群

$V'''_1 = (1.1223, 0.6363, 1.2303)$,　$V'''_2 = (0.5631, 0.6812, 1.1614)$,
$V'''_3 = (0.6068, 0.6540, 1.4781)$,　$V'''_4 = (1.0155, 0.6722, 1.2051)$,

$V'''_5 = (0.9167,0.2930,0.3297)$, $V'''_6 = (0.6818,0.9477,1.2685)$,
$V'''_7 = (0.9167,0.2930,0.3297)$, $V'''_8 = (0.8523,0.9683,1.4477)$,
$V'''_9 = (1.3047,0.4163,0.4673)$, $V'''_{10} = (0.9263,1.5153,0.0503)$,
$V'''_{11} = (0.0823,0.1867,1.1217)$, $V'''_{12} = (0.6088,0.8088,1.1092)$,
$V'''_{13} = (0.9810,1.2702,0.3794)$, $V'''_{14} = (0.9263,1.5153,0.0503)$,
$V'''_{15} = (0.8236,0.7808,1.0040)$, $V'''_{16} = (1.6400,0.6222,0.4287)$,
$V'''_{17} = (0.8523,0.9683,1.4477)$, $V'''_{18} = (0.9263,1.5153,0.0503)$,
$V'''_{19} = (0.7517,1.2184,0.3953)$, $V'''_{20} = (0.1313,0.8217,1.4523)$,
$V'''_{21} = (0.9263,1.5153,0.0503)$, $V'''_{22} = (1.3047,0.4163,0.4673)$,
$V'''_{23} = (1.1223,0.6363,1.2303)$, $V'''_{24} = (1.6243,1.0153,0.5573)$,
$V'''_{25} = (1.1910,0.1460,0.5890)$, $V'''_{26} = (0.9907,0.8447,0.9030)$,
$V'''_{27} = (0.1517,1.0047,0.5590)$, $V'''_{28} = (1.0677,0.8814,0.9013)$,
$V'''_{29} = (0.0490,1.7077,0.2813)$, $V'''_{30} = (1.6243,1.0153,0.5573)$.

类似于交叉过程，经过计算机运行，有以下染色体被选择作为父代用来进行变异操作：

$$V'''_1, V'''_3, V'''_4, V'''_7, V'''_9, V'''_{10}, V'''_{13}, V'''_{14}, V'''_{15},$$
$$V'''_{16}, V'''_{17}, V'''_{18}, V'''_{19}, V'''_{21}, V'''_{22}, V'''_{23}, V'''_{24}.$$

在 $\mathbf{R}^3$ 中，随机产生一个方向 $\boldsymbol{d} = (0.1130, -0.8072, 0.1279)$，在这个方向上，对染色体 V'''_1 进行变异. 不妨取 M 为 10，检验

$$V'''_1 + M\boldsymbol{d} = (2.2523, -7.4357, 2.5093),$$

知其不可行. 从 0 到 M 之间产生的一个随机数为 0.2345，置 $M = 0.2345$. 检验函数显示

$$V'''_1 + M\boldsymbol{d} = (1.1488, 0.4470, 1.2333)$$

是可行的. 所以用(1.1488,0.4470,1.2333)代替

$$V'''_1 = (1.1223, 0.6363, 1.2303).$$

经过变异操作，得到种群

$V''''_1 = (1.1488,0.4470,1.2333)$, $V''''_2 = (0.5631,0.6812,1.1614)$,
$V''''_3 = (0.1701,0.6540,1.4550)$, $V''''_4 = (0.9978,0.6722,1.2077)$,
$V''''_5 = (0.9167,0.2930,0.3297)$, $V''''_6 = (0.6818,0.9477,1.2685)$,
$V''''_7 = (0.9811,0.2930,0.1346)$, $V''''_8 = (0.8523,0.9683,1.4477)$,
$V''''_9 = (1.3047,0.4163,0.4285)$, $V''''_{10} = (0.8192,1.5153,0.0503)$,
$V''''_{11} = (0.0823,0.1867,1.1217)$, $V''''_{12} = (0.6088,0.8088,1.1092)$,
$V''''_{13} = (0.9810,1.2702,0.6591)$, $V''''_{14} = (0.1878,1.5153,0.0503)$,
$V''''_{15} = (0.8236,0.7808,0.7472)$, $V''''_{16} = (1.7911,0.6222,0.3176)$,
$V''''_{17} = (0.6160,0.9683,1.4477)$, $V''''_{18} = (1.0359,1.5153,0.0503)$,
$V''''_{19} = (0.7517,1.2268,0.3953)$, $V''''_{20} = (0.1313,0.8217,1.4523)$,

$V''''_{21} = (0.9263, 1.7268, 0.3645)$, $V''''_{22} = (1.2524, 0.4163, 0.4673)$,
$V''''_{23} = (1.1223, 0.6363, 1.2303)$, $V''''_{24} = (1.6243, 0.8745, 0.5573)$,
$V''''_{25} = (1.1910, 0.1460, 0.5890)$, $V''''_{26} = (0.9907, 0.8447, 0.9030)$,
$V''''_{27} = (0.1517, 1.0047, 0.5590)$, $V''''_{28} = (1.0677, 0.8814, 0.9013)$,
$V''''_{29} = (0.0490, 1.7077, 0.2813)$, $V''''_{30} = (1.6243, 1.0153, 0.5573)$.

到此，已经得到新一代的种群.

经过 150 代之后，得到最好的解为

$$x^* = (0.8597, 0.5273, 1.3245),$$

其目标值为 $f(x^*) = 0.1537$，已接近已知的最优解，所花费的 CPU 时间为 4.1 秒. 进化过程由图 2-2 给出.

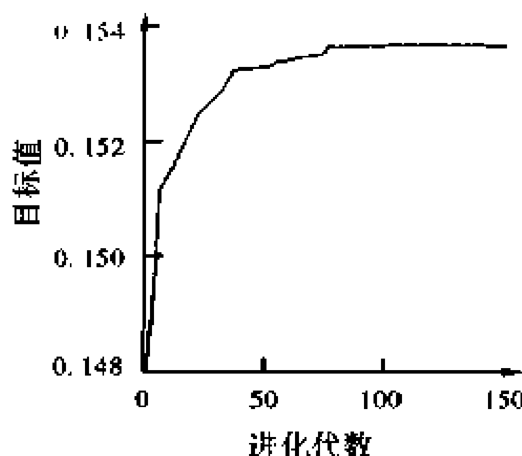


图 2-2

例 2 目标规划. 考虑非线性目标规划模型

$$\begin{cases} \text{lexmin}\ \{d_1^-, d_2^-, d_3^+, d_4^- + d_4^+\}, \\ \text{s.t.}\ \sum_{i=1}^{5} x_i \sin(i\pi \cdot x_i) + d_1^- - d_1^+ = 18, \\ \sum_{i=1}^{4} x_i \sin(i\pi \cdot x_i) + d_2^- - d_2^+ = 15, \\ \sum_{i=1}^{3} x_i \sin(i\pi \cdot x_i) + d_3^- - d_3^+ = 10, \\ \sum_{i=1}^{2} x_i \sin(i\pi \cdot x_i) + d_4^- - d_4^+ = 0, \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \leqslant 100. \end{cases}$$

此模型是很复杂的，目标函数不但是多峰的，而且是高度非线性的. 传统方法无法处理这类非线性目标规划，但遗传算法确是有效的.

目标规划的目标函数的一般形式为

$$\sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-),$$

其中 P_j 为优先因子，表示各个目标的相对重要性，对所有的 j，有 $P_j \gg P_{j+1}$. 但它并不适合作为目标规划的评价函数，因为对优先因子，仅有信息 $P_j \gg P_{j+1}$. 事实上，对染色体，有如下的序的关系：对任意两个染色体，如果在较高的优先级中目标值相等，则在当前优先级中，较小的目标值对应的比较好. 这种关系是可行集上的一个序，从而可以按这个序重排这些染色体. 如果两个染色体具有相同的目标值，则可以随机重排这两个染色体.

除了在每个种群中按上述序的关系重排染色体这一点外，目标规划的进化过程与单目标规划的进化过程没有什么区别.

经过 6 000 次进化后，遗传算法给出的解为

$$x = (2.2710, 1.4520, -6.8352, 6.1263, 2.9016),$$

它满足前三个目标，但最后一个目标值为 2.1397. 虽然不知道这个问题的精确最优解，但通过大量的实验，没有发现比这更好的解，所以我们对这个结论是满意的. 进化 6 000 代，所花费的 CPU 时间是 68.8 秒，进化过程如图 2-3.

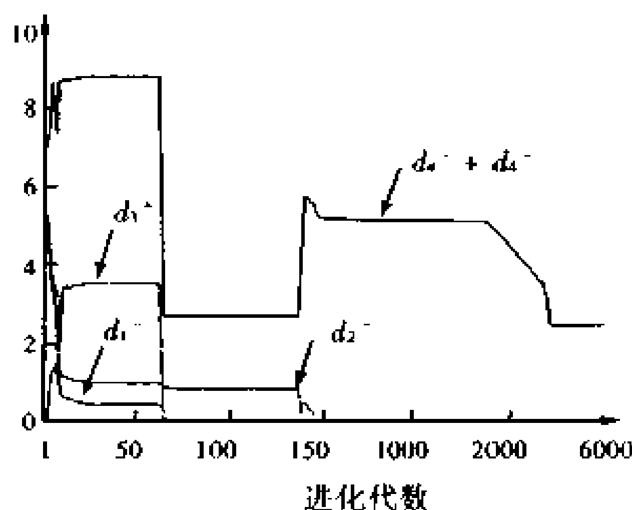


图 2-3

例 3　多目标规划. 虽然我们可能对多目标规划的所有有效解感兴趣，但不一定非要找出所有的有效解. 解决多目标优化问题的目的在于在实际过程中执行这个解，而同一时刻不可能执行所有的解，因此，一个满意的解对某些实际问题已经足够了.

无论一个问题有多少个目标，合理的假设是决策者能够对任意两个可能解进行比较，即指出它们的好坏. 可以采用下面的方法之一评估可能的解. 例如，可采用加权和方法，对 m 个目标函数 $f_i, i = 1,2,\cdots,m$ 进行合并，令

$$f(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x) + \cdots + \lambda_m f_m(x),$$

其中 $\lambda_1, \lambda_2, \cdots, \lambda_m$ 是权重因子，$0 \leqslant \lambda_i \leqslant 1, i = 1,2,\cdots,m$，且 $\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1$. 又如，可采用距离函数法，根据理想目标向量 $(f_1^0, f_2^0, \cdots, f_m^0)$ 把 m 个目标函数合并成

$$f(\boldsymbol{x}) = \Big(\sum_{i=1}^{m} | f_i(\boldsymbol{x}) - f_i^0 |^\lambda\Big)^{\frac{1}{\lambda}}.$$

其中 λ 是正的参数.采用上述方法可以对多目标规划的任意一组可行解由好到坏进行重排.

现考虑非凸集合上的多目标规划

$$\begin{cases} \max f_1(\boldsymbol{x}) = x_1^2 + x_2^2, \\ \max f_2(\boldsymbol{x}) = \dfrac{x_3}{(1 + x_1 + x_2)}, \\ \text{s.t. } x_1^2 + x_2^2 + x_3^2 \geq 1, \\ \qquad x_1^2 + x_2^2 + x_3^2 \leq 4, \\ \qquad x_1, x_2, x_3 > 0. \end{cases}$$

如果使用加权和方法重排染色体,例如,对目标函数 $f_1(\boldsymbol{x})$ 和 $f_2(\boldsymbol{x})$ 分别给予权重0.4和0.6.经过2 000代以后,遗传算法给出的最优解为 $\boldsymbol{x}^* = (0.0000, 1.9837, 0.2545)$,其对应的目标值是

$$f_1(\boldsymbol{x}^*) = 3.9352, \qquad f_2(\boldsymbol{x}^*) = 0.0853.$$

所花费的 CPU 时间为32.4秒.

3　几个应用问题

本节给出几个应用遗传算法能够求解的典型问题,包括运输问题、流动推销员问题、统计问题、泛函优化问题和障碍区域最短路问题,其目的是让读者进一步了解应如何对问题的解进行编码以及如何设计遗传算法程序.

3.1　运输问题

运输问题是个组合优化问题,它是为将多处资源运到多处目的地而寻找一个最低费用运输方案的问题.自20世纪40年代起,就已经开始了运输问题的研究.

设有 m 处资源,其资源拥有量分别为 $a_1, a_2, \cdots, a_m$,又有 n 处目的地,其资源需求量分别为 $b_1, b_2, \cdots, b_n$.设 x_{ij} 分别为第 i 处资源运到第 j 处目的地的量,$i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$.首先,从任何一处调出的资源量都不能超过其拥有量,于是有约束条件

$$x_{i1} + x_{i2} + \cdots + x_{in} \leq a_i, \quad i = 1, 2, \cdots, m.$$

其次,每一处目的地的需求量都应得到满足,于是又有约束条件

$$x_{1j} + x_{2j} + \cdots + x_{mj} \geq b_j, \; j = 1, 2, \cdots, n.$$

另一个明显的约束是运输量不能为负值,即

$$x_{ij} \geq 0, \; i = 1, 2, \cdots, m, \quad j = 1, 2, \cdots, n.$$

设第 i 处资源到第 j 处目的地的运输费用为 $f_{ij}(x_{ij})$, $i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$,

则总运输费用为 $\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}(x_{ij})$. 因此,该运输问题建模如下,

$$\begin{cases}\min\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}(x_{ij}),\\ \text{s.t.}\sum_{j=1}^{n}x_{ij}\leqslant a_i,\quad i=1,2,\cdots,m,\\ \quad\sum_{i=1}^{m}x_{ij}\geqslant b_j,\quad j=1,2,\cdots,n,\\ \quad x_{ij}\geqslant 0,\quad i=1,2,\cdots,m,j=1,2,\cdots,n.\end{cases}$$

也就是说,为确定最低费用运输方案,只要求出上述问题的最优解就可以了.

这里只考虑如下的平衡运输问题:

$$\begin{cases}\min\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}(x_{ij}),\\ \text{s.t.}\sum_{j=1}^{n}x_{ij}=a_i,\quad i=1,2,\cdots,m,\\ \quad\sum_{i=1}^{m}x_{ij}=b_j,\quad j=1,2,\cdots,n,\\ \quad x_{ij}\geqslant 0,\quad i=1,2,\cdots,m,j=1,2,\cdots,n.\end{cases}$$

为了保证该运输问题有可行解,必须满足平衡条件

$$\sum_{i=1}^{m}a_i=\sum_{j=1}^{n}b_j.$$

对于如此的运输问题,也许解的最自然表示是用一个 $m\times n$ 矩阵表示,即染色体表示为

$$V=\begin{bmatrix}x_{11}&x_{12}&\cdots&x_{1n}\\x_{21}&x_{22}&\cdots&x_{2n}\\\vdots&\vdots&&\vdots\\x_{m1}&x_{m2}&\cdots&x_{mn}\end{bmatrix}.$$

首先,须按如下方式随机生成初始染色体种群:

步 1:置 $I=\{1,2,\cdots,m\}$, $J=\{1,2,\cdots,n\}$.

步 2:在 I 和 J 中分别产生一个随机数,记为 i 和 j.

步 3:置 $x_{ij}=\min\{a_i,b_j\}$.

步 4:置 $a_i\leftarrow a_i-x_{ij}$, $b_j\leftarrow b_j-x_{ij}$, $I\leftarrow I/\{i\}$, $J\leftarrow J/\{j\}$.

步 5:若 I 与 J 非空,则回到步 2.

由此产生的染色体肯定是可行解.重复以上过程 N 次,即可得到初始种群.

交叉操作可采用前述的算术交叉,且产生的后代也一定是可行的.

变异操作可先给定两个整数 p 和 $q(2\leqslant p\leqslant m,2\leqslant q\leqslant n)$. 设 $\{i_1,i_2,\cdots,i_p\}$ 和 $\{j_1,j_2,\cdots,j_q\}$ 分别为 I 和 J 的两个子集.可按下述方式建立一个染色体 V 的 $p\times$

q 子矩阵 $W = [w_{ij}]$：一个 V 中元素 x_{ij} 在 W 中，当且仅当 $i \in \{i_1, i_2, \cdots, i_p\}$ 且 $j \in \{j_1, j_2, \cdots, j_q\}$. 对于子矩阵 W，可用初始化方法重新随机设定 W 的元素，只是此时

$$a_i = \sum_{j \in \{j_1, j_2, \cdots, j_q\}} x_{ij}, \quad 1 \leqslant i \leqslant p,$$

$$b_j = \sum_{i \in \{i_1, i_2, \cdots, i_p\}} x_{ij}, \quad 1 \leqslant j \leqslant q.$$

然后，将 W 的元素替换 V 中相应的元素形成一个后代 V'，且该染色体 V' 也是可行的.

将这些融合到一般的遗传过程中，就可组成一个求解平衡运输问题的遗传算法.

3.2 流动推销员问题

这也是一个典型的组合优化问题. 流动推销员问题是为一个推销员漫游 n 个城市寻找一条最短路. 现已有人用分枝定界法求解了 2000 多个城市的流动推销员问题. 虽然已有很多学者应用遗传算法求解流动推销员问题，但遗传算法还远远不能解决如此大规模的问题.

应用遗传算法求解流动推销员问题的关键是旅行路线的编码方式以及如何设计合适的遗传操作以保证推销路线的合法性.

最自然的推销路线表示是城市的访问次序. 设有 7 个城市，分别标号为 1,2, …,7. 当一个推销路线为 5 → 3 → 6 → 5 → 4 → 2 → 7 时，其染色体表达方式为(5, 3,6,1,4,2,7). 然而，对于这种表达方式，似乎所有的遗传算子都会产生非法的推销路线. 于是，为应用这种表达方式，我们必须采用一些修复策略以保证新的染色体所对应的推销路线的合法性. 例如，当一个染色体经过交叉和变异后，设其重复的城市有 1,3,4，而丢掉的城市有 2,5,7. 则去掉重复的城市 1,3,4 而用城市 2,5,7 代替. 这样，一个非法的推销路线就被修复成一个合法的推销路线.

另一种表达方式是用一组介于 0 与 1 之间的随机数组成的向量表示. 例如，对于 7 个城市的流动推销员问题，若染色体为 (0.43,0.77,0.21,0.58,0.14,0.29, 0.92)，其中第 i 个位置代表 i 个城市，而推销路线则按城市对应值的升序排列. 于是该染色体所代表的推销路线为 5—3—6—1—4—2—7. 这种方式定义的染色体容易保证推销路线的合法性. 只要每个分量介于 0 与 1 之间，则其对应的推销路线必是合法的.

现在考虑一个 11 个城市的流动推销员问题，这 11 个城市的标号分别为 0,1,2, 3,…,10，它们之间的距离矩阵如表 3-1 所示.

在表 3-1 中，(11 × 11) 维矩阵的第 i 行第 j 列元素表示第 i 个城市到第 j 个城市的距离，其中 $i = 0,1,2,\cdots,10, j = 0,1,2,\cdots,10$. 注意，第 i 个城市到第 j 个城市的距离不一定等于第 j 个城市到第 i 个城市的距离.

假设某个流动推销员由城市 0 出发，漫游其余 10 个城市，最后回到城市 0. 此时的流动推销员问题是确定一条最短的推销践线. 为应用遗传算法求解该问题，定义

表3-1

城市	0	1	2	3	4	5	6	7	8	9	10
0	—	3	93	66	13	100	25	33	9	57	19
1	24	—	33	77	42	21	16	45	34	21	109
2	45	107	—	81	36	16	4	28	25	37	62
3	139	90	80	—	56	7	44	56	20	91	75
4	18	64	188	33	—	11	25	96	5	57	43
5	3	88	18	46	92	—	55	33	20	91	7
6	44	26	33	27	84	39	—	101	9	72	36
7	11	39	24	98	103	76	54	—	50	63	99
8	77	82	67	19	30	42	56	9	—	88	28
9	12	133	32	69	21	52	87	66	43	—	55
10	92	32	81	73	44	24	64	15	77	9	—

种群规模为20,然后随机生成20个从1到10的自然数排列,每个排列为一个初始染色体,每个染色体对应一个推销路线.例如,染色体(1,2,3,4,5,6,7,8,9,10)对应的推销路线为0→1→2→3→4→5→6→7→8→9→10→0.另外,可按如下方式随机生成一个染色体,即从1到10的自然数排列:首先定义一个10维数组$T[10]$,并置$T[i]=i+1, i=0,1,2,\cdots,9$.在$C$语言库函数中,对于任何的正整数$k$,子函数random($k$)将产生0到$k-1$之间的随机整数.定义$n=10$.利用该函数产生一个0到$n-1$之间的整数,记为$m$,则$T[m]$即为染色体的第一基因.将数组$T$的第$m$个元素$T[m]$拿掉,再将第$m$位后的元素依次左移一位,形成一个新数组,仍记其为$T$.将$n$减小为9,重复以上过程,得到的$T[m]$即为染色体的第二个基因.再次减小$n$的值,并重复以上过程,直到生成一个完整的染色体.

计算这20个染色体所对应的推销路线的实际距离,并以此为依据将染色体由好到坏进行重排.再利用基于序的评价函数确定其适应度.旋转20次赌轮,得到新一代的染色体.

对于新一代的染色体,首先对其进行交叉操作.这里定义交叉概率为0.3.将选择的染色体两两配对后进行交叉.随机抽取染色体的一段基因,然后与另一个染色体相应的基因段互换.于是产生两个新的染色体.这两个染色体不一定对应合法路线,此时,可以采取修复策略使它们成为合法染色体.将这两个染色体取代种群中的父染色体.

接下来进行变异.定义变异概率为0.2.采取如下两种方式进行变异.第一种将染色体随机抽取一段基因进行随机重排,进而变异出新的染色体.第二种在染色体中随机抽取两段基因并互换位置.

执行以上进化过程100代,我们将最好的染色体对应的推销路线视为最优解.事实上,运行遗传算法100代得到的最佳染色体为(1,2,6,3,5,10,9,4,8,7),对应

的旅行路线为0—1—2—6—3—5—10—9—4—8—7—0，而总推销距离为 $3+33+4+27+7+7+9+21+5+9+11=136$.

3.3 统计问题

一类统计问题是在保序约束下根据极大似然估计原理去确定概率分布的参数. 令 X_i 是具有绝对连续概率分布 $F\left(\frac{x-\theta_i}{\sigma_i}\right)$ 的随机变量，x_{ij} 表示观察值，θ_i 和 σ_i 是未知参数，$j=1,2,\cdots,n_i$，$i=1,2,\cdots n$. 已知参数 θ_i 和 σ_i 具有一定的偏序关系，如：

简单序：$\theta_1\leqslant\theta_2\leqslant\cdots\leqslant\theta_n,\sigma_1\leqslant\sigma_2\leqslant\cdots\leqslant\sigma_n$，

伞形序：$\theta_1\leqslant\cdots\leqslant\theta_h\geqslant\theta_{h+1}\geqslant\cdots\geqslant\theta_n,\sigma_1\leqslant\cdots\leqslant\sigma_h\geqslant\sigma_{h+1}\geqslant\cdots\geqslant\sigma_n$，

树状序：$\theta_1\leqslant\theta_j,\sigma_1\leqslant\sigma_j,j=2,3,\cdots,n$，

等等.

现在的问题是在给定的观察值 $x_{ij},j=1,2,\cdots,n_i,i=1,2,\cdots,n$ 及偏序关系下确定概率分布中的未知参数 θ_i 和 $\sigma_i,i=1,2,\cdots,n$. 首先，需要建立其数学模型.

例如，当 $F\left(\frac{x-\theta_i}{\sigma_i}\right)$ 为正态分布函数，且偏序关系为 $\theta_1\leqslant\theta_2\leqslant\cdots\leqslant\theta_n,\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n$ 时，统计参数估计问题即为求解如下的最优化问题：

$$\begin{cases}\max\sum_{i=1}^{n}\left[-\frac{n_i}{2}\mathrm{lb}\sigma_i^2-\frac{1}{2\sigma_i}\sum_{j=1}^{n_i}(x_{ij}-\theta_i)^2\right],\\ \text{s.t.}\ \theta_1\leqslant\theta_2\leqslant\cdots\leqslant\theta_n,\\ \qquad\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n.\end{cases}$$

又如，当 $F\left(\frac{x-\theta_i}{\sigma_i}\right)$ 为拉普拉斯分布函数，且偏序关系为 $\theta_1\leqslant\cdots\leqslant\theta_h\geqslant\theta_{h+1}\geqslant\cdots\geqslant\theta_n,\sigma_1\geqslant\cdots\geqslant\sigma_h\leqslant\sigma_{h+1}\leqslant\cdots\leqslant\sigma_n$ 时，统计参数估计问题可化为如下的优化问题，

$$\begin{cases}\max\sum_{i=1}^{n}\left[-n_i\mathrm{lb}\sigma_i-\frac{1}{\sigma_i}\sum_{j=1}^{n_i}|x_{ij}-\theta_i|\right],\\ \text{s.t.}\quad\theta_1\leqslant\cdots\leqslant\theta_h\geqslant\theta_{h+1}\geqslant\cdots\geqslant\theta_n,\\ \qquad\sigma_1\geqslant\cdots\geqslant\sigma_h\leqslant\sigma_{h+1}\leqslant\cdots\leqslant\sigma_n.\end{cases}$$

求解这类问题，对于经典算法来说，并不是一件容易的事情. 然而，遗传算法却能胜任求解这类问题的任务.

用 $2\times n$ 阶矩阵

$$V=\begin{bmatrix}x_1 & x_2 & \cdots & x_n\\ y_1 & y_2 & \cdots & y_n\end{bmatrix}$$

作为染色体表示参数的估计值. 一个关键问题是建立起染色体 V 与参数 θ_i 和 σ_i 的映射关系，且该映射关系能保持 θ_i 和 σ_i 的序关系.

对于简单序关系 $\theta_1 \leqslant \theta_2 \leqslant \cdots \leqslant \theta_n$，可以将序列 $\{x_1, x_2, \cdots, x_n\}$ 由小到大重排，然后令 $\theta_i = x_i, i = 1,2,\cdots,n$. 序列 $\{y_1, y_2, \cdots, y_n\}$ 可按同样方式映射到 $\{\sigma_1, \sigma_2, \cdots, \sigma_n\}$.

对于伞形序 $\theta_1 \leqslant \cdots \leqslant \theta_h \geqslant \theta_{h+1} \geqslant \cdots \geqslant \theta_n$，首先找出 $\{x_1, x_2, \cdots, x_n\}$ 中最大元素，例如 x_m，然后交换 x_m 与 x_h 的值. 再将 $\{x_1, x_2, \cdots, x_h\}$ 由小到大重排，将 $\{x_{h+1}, x_{h+2}, \cdots, x_n\}$ 由大到小重排，进而令 $\theta_i = x_i, i = 1,2,\cdots,n$ 即得.

对于树形序 $\theta_1 \leqslant \theta_j, j = 2,3,\cdots,n$，可找出 $\{x_1, x_2, \cdots, x_n\}$ 中最小元素，如 x_m，然后交换 x_1 与 x_m 的值，再令 $\theta_i = x_i, i = 1,2,\cdots,n$ 即得.

将如上的映射关系应用于进化过程，即可得到确定统计参数的遗传算法.

例如，为了探索施肥量与粮食亩产量之间的关系，共做了4组实验. 记第 i 组实验的施肥量为 p_i，第 i 组第 j 次实验得到的粮食亩产量为 x_{ij}，其中 $i = 1,2,3,4$. 实验数据如表 3-2 所示.

表 3-2

实验	$p_1 = 0$	$p_2 = 5$	$p_3 = 7$	$p_4 = 10$
亩产量	$x_{11} = 300$	$x_{21} = 340$	$x_{31} = 310$	$x_{41} = 370$
	$x_{12} = 360$	$x_{22} = 380$	$x_{32} = 365$	$x_{42} = 375$
		$x_{23} = 330$	$x_{33} = 405$	$x_{43} = 345$
			$x_{34} = 305$	$x_{44} = 328$
				$x_{45} = 410$

假设这些数据分别来自正态总体 $N(\mu_i, \sigma_i^2)$，$i = 1,2,3,4$，其中参数 μ_i 和 σ_i 是未知的，但应满足如下的序关系：

$$\mu_1 \leqslant \mu_2 \leqslant \mu_3 \leqslant \mu_4,$$

且

$$\sigma_1^2 \geqslant \sigma_2^2 \geqslant \sigma_3^2 \geqslant \sigma_4^2 > 0.$$

为了估计 μ_i 和 σ_i^2 的值，可以用极大似然估计法，这等价于求解极大化问题

$$\begin{cases} \max \sum_{i=1}^{4} \left[-\frac{n_i}{2} \mathrm{lb} \sigma_i^2 - \frac{1}{2\sigma_i^2} \sum_{j=1}^{n_i} (x_{ij} - \mu_i)^2 \right], \\ \text{s.t.} \quad \mu_1 \leqslant \mu_2 \leqslant \mu_3 \leqslant \mu_4, \\ \qquad \sigma_1^2 \geqslant \sigma_2^2 \geqslant \sigma_3^2 \geqslant \sigma_4^2 > 0, \end{cases}$$

其中 x_{ij} 是表 3-2 中所示的亩产量，$i = 1,2,3,4, j = 1,2,\cdots,n_i$.

应用前述的编码方式，运行遗传算法 3000 代，得到的概率分布的均值与方差为

$$(\mu_1, \mu_2, \mu_3, \mu_4) = (330.03, 347.87, 347.87, 365.62),$$
$$(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2) = (2650.01, 2650.01, 2650.01, 1801.62).$$

3.4 泛函最优化

泛函最优化是寻找一个函数而不是一个向量去优化一个泛函.泛函优化问题可以用变分原理或其它数学方法处理,但这类数学方法只对特殊的泛函优化问题有效.

下面将看到这类问题也可以通过遗传算法求出近似解.例如,

$$\begin{cases} \min\limits_{\varphi}\int_0^1[\sin^3 x + \varphi(x)^2]\mathrm{d}x, \\ \text{s.t.}\ \varphi(x) \in C[0,1], \\ \qquad \varphi(x) \geqslant 0, \\ \qquad \int_0^1 x[\varphi(x)]^{4/3}\mathrm{d}x = 1, \end{cases}$$

其中 $C[0,1]$ 表示区间$[0,1]$上的所有连续函数集合.现设计的计算程序将用 5 个点所代表的折线去近似该问题的解 $\varphi(x)$,故可用如下的矩阵表示 5 个点的染色体

$$V = \begin{bmatrix} x_1 & x_2 & \cdots & x_5 \\ y_1 & y_2 & \cdots & y_5 \end{bmatrix},$$

其中 $x_1 = 0, x_5 = 1, x_1 < x_2 < \cdots < x_5$ 且 $y_i \geqslant 0, i = 1,2,\cdots,5$.

为了保证产生的染色体是可行的,即满足 $\{\varphi(x) \in C[0,1] \mid \varphi(x) \geqslant 0, \int_0^1 x[\varphi(x)]^{4/3}\mathrm{d}x = 1\}$,将按如下方式调整每一个新产生的染色体:首先,计算积分

$$A = \int_0^1 x[\varphi(x)]^{4/3}\mathrm{d}x,$$

其中 $\varphi(x)$ 是由原染色体 V(此时它可能是不可行的)确定的折线.然后,用 $A^{-3/4}y_i$ 分别代替 $y_i, i = 1,2,\cdots,5$.于是,一个新的染色体产生了,仍记其为 V.容易验证,只要 $\varphi(x)$ 是由新的染色体确定的折线,则必有

$$\int_0^1 x[\varphi(x)]^{4/3}\mathrm{d}x = 1.$$

已知该问题的最优解是

$$\varphi^*(x) = (2x)^{3/2},$$

且最优目标值是 $J(\varphi^*) = 2.1790$.

遗传算法运行 200 代得到的最优解 φ^* 是连接以下五个点的折线:

(0.0000,0.0169),(0.2506,0.3347),(0.4209,0.8346),
(0.6456,1.4288),(1.0000,2.7928),

而它所对应的泛函值为 2.1792.该目标值已很接近实际最优值.

3.5 障碍区域最短路问题

在一个存在已知障碍物的区域,下面的问题是寻找一条绕过所有障碍物连接

两个固定点的最短曲线.

图3-3表示一个10×10区域,其中障碍物为三个长方形,

$$R_1 = \{(x,y) \mid 1 \leqslant x \leqslant 2, 4 \leqslant y \leqslant 10\},$$
$$R_2 = \{(x,y) \mid 3 \leqslant x \leqslant 4, 0 \leqslant y \leqslant 6\},$$
$$R_3 = \{(x,y) \mid 5 \leqslant x \leqslant 7, 4 \leqslant y \leqslant 8\},$$

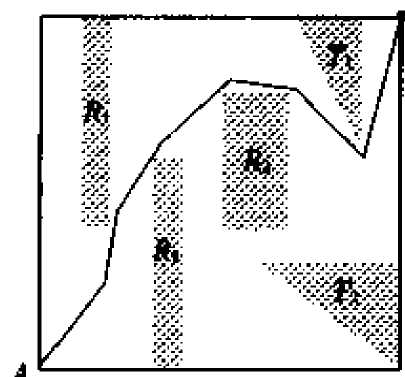


图 3-3

和两个三角形

$$T_1 = \{(x,y) \mid 7 \leqslant x \leqslant 9, 24 - 2x \leqslant y \leqslant 10\},$$
$$T_2 = \{(x,y) \mid 6 \leqslant x \leqslant 10, -\frac{3}{4}(x-10) \leqslant y \leqslant 3\}.$$

应用遗传算法搜索连接 $A = (0,0)$ 和 $B = (10,10)$ 两点的最短路.为了保证染色体的可行性,将采用启发式方法:首先,由点 A 出发,根据随机产生的方向和步长移动到新的一点.重复这个过程直到新的一点落到与 B 点能直线连通的区域.由此得到一条连接 A 和 B 两点的折线,且该折线可由这一系列的折点表示,设这些点为 $(x_i, y_i), i = 1,2,\cdots,m$,其中,$x_1 = 0, y_1 = 0, x_m = 10, y_m = 10$.容易得出该折线长为

$$\sum_{i=1}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}.$$

将染色体表示为

$$V = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ y_1 & y_2 & \cdots & y_m \end{bmatrix},$$

其中 m 是可变的.通过以上的启发式方法得到初始种群.在以后的遗传操作中,须注意新产生的染色体的可行性.运行遗传算法3000代得到的最短路为由以下8个点确定的折线,

(0,0),　(1.7728,2.4215),　(2.1306,4.4594),　(3.1875,6.2167)
(5.2745,8.2190),　(7.1283,7.9803),　(9.0000,6.0000),　(10,10),

而路径长度为18.73.

参 考 文 献

1　Bäck T. Evolutionary algorithms in theory and practice. New York: Oxford University Press, 1995.

2　Forgel DB. Evolution computation: toward a new philosophy of machine intelligence. Piscataway: IEEE Press, 1995.

3　Goldberg DE. Genetic algorithms in search, optimization and machine Learning. Reading MA: Addison-Wesley, 1989.

4　Holland JH. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michi-

gan Press, 1975.

5 Koza JR. Genetic programming. Cambridge: MIT Press, 1992.

6 Koza JR. Genetic programming, II. Cambridge: MIT Press, 1994.

7 Liu B. Uncertain Programming. New York: Wiley, 1999.

8 Michalewicz Z. Genetic algorithms + data structures = evolution programs. 3nd ed., New York: Springer-Verlag, 1996.

9 Mitchell M. An Introduction to Genetic Algorithms. Cambridge: MIT Press, 1996.

10 刘宝碇,赵瑞清.随机规划与模糊规划.北京:清华大学出版社,1998.

·计算机数学卷·

第 15 篇

模拟退火算法

编　者　李元香
审校者　康立山

目　　录

引　言

大量的组合优化问题，至今还不存在有效的多项式时间求解算法，退而求其近优解便是必然的途径，一种启发式算法(heuristic algorithms)便应运而生. 模拟退火算法(simulated annealing algorithm)便是它的代表. 模拟退火算法是受金属的冷却过程的启发，最早由米特罗波利斯(Metropolis)于1953年提出. 由于它的灵活有效，研究者将它的应用范围日益扩大，对许多著名的NP难题都提出了它的有效办法，所以成为今日备受关注的一类近似算法，本篇介绍它的基本思想及其应用.

1　物理学中的模拟退火

1.1　退火的概念

模拟退火算法是模拟金属固体热浴后的冷却，即退火过程. 将某种金属材料加热到一定的温度并持续一定时间后缓慢冷却，这就是退火. 物理学原理告诉我们，冷却后的固体结构依赖于冷却速度，缓慢冷却产生稳定的晶体结构，而快速冷却会导致晶体中包含一些不稳定的因素. 运用统计物理学的方法来分析，金属的退火过程可以用一个粒子系统进行模拟. 将金属物质看成是一个粒子系统(实际上物质就是由分子组成的)，金属的退火过程就是该粒子系统的热动力学演变过程. 在这个演变过程中系统的能量是一个关键的热力学量，它与系统的状态密切相关.

1.2　统计物理学基础

统计物理从微观的粒子运动层次上研究物理系统，研究对象是微观粒子系统，目的是为了揭示宏观物理现象背后的微观动力学规律.

(1) **宏观态**　一个物理系统的宏观态是指系统的热力学状态，用体积、温度、压力和能量等宏观上可观测的一些物理量来表征.

(2) **微观态**　一个物理系统的微观态是指系统的动力学状态，用微观粒子的位置与运动状态来表征. 设一个系统由 N 个粒子组成，每个粒子有 s 个自由度，则表征系统的微观态需要 Ns 个广义坐标和 Ns 个广义动量.

(3) **相空间**　表示系统的 $2Ns$ 个广义坐标和广义动量组成的 $2Ns$ 维空间称为系统的相空间. 相空间中的一点代表系统的一个微观态，称为代表点. 通常一个系统的宏观态对应于许多微观态，宏观态的特性是所有这些微观态特性的平均.

(4) **系综**　系综是一群代表点的集合，这些代表点代表着不同的微观态，但对

应于系统的同一个宏观态.设想同时考虑极大数目的 M 个类似系统,每个系统都包含 N 个相同粒子,都在同样的外界条件下,具有相同的宏观性质,但微观态不同,代表点在相空间的位置也就不同,M 个代表点在 $2Ns$ 维相空间中的分布也会有疏有密.

(5) **系综分布函数** 设相空间用 Γ 表示,取小体积元 $\Delta\Gamma$,其中有 ΔM 个代表点,称

$$D = \lim_{\Delta\Gamma\to 0}\frac{\Delta M}{\Delta\Gamma}$$

为相密度(代表点的密度).恰好有一个代表点落在 $\Delta\Gamma$ 内的概率为 $\Delta w = \frac{D\Delta\Gamma}{M}$.系综分布函数定义为

$$\rho = \lim_{\Delta\Gamma\to 0}\frac{\Delta w}{\Delta\Gamma}. \tag{1-1}$$

显然有 $\rho = \frac{D}{M}$,且

$$\int_{\Gamma}\rho(q,p,t)\mathrm{d}\Gamma = 1, \tag{1-2}$$

其中 q 表示 Γ 中的广义坐标,p 表示广义动量,t 是时间变量.(1-2) 式表明 ρ 是一个概率密度函数.

(6) **先验等几率假设** 考虑 M 个这样的系统,它们具有确定的粒子数和确定的体积,系统基本孤立,能量只能在 E 到 $E+\Delta E$ 之间变动,这 M 个系统组成的系综称为**微正则系综**.先验等几率假设是:微正则系综中任一微观态出现的概率相等,即系综分布函数为一均匀分布:

$$\rho = \frac{1}{M}. \tag{1-3}$$

先验等几率假设是平衡态统计物理中一个最基本的原理.

(7) **玻耳兹曼(Boltzmann) 分布** 考虑这样一些系统,它们具有确定的粒子数和确定的体积,系统与一恒温热源相接触而平衡,能量可变,这些系统组成的系综称为**正则系综**.由微正则系综的分布函数(1-3) 式可以导出正则系综的分布函数:

$$\rho_i = \frac{1}{Z}\exp\left(\frac{-E_i}{kT}\right), \tag{1-4}$$

其中 E_i 是微观态 i 的能量,T 是系统的(绝对)温度,k 称为玻耳兹曼常数,Z 称为配分函数,

$$Z = \sum_i \exp\left(\frac{-E_i}{kT}\right),$$

$\sum_i$ 表示对所有微观态求和.

分布函数(1-4) 就是著名的玻耳兹曼分布,它基本上能描述真实物理系统的性质.(1-4) 式表明系统处于能量最低的微观态的概率最大.

1.3 米特罗波利斯准则

物理系统总是倾向于能量最低态,而粒子的热运动又妨碍它准确处于能量最低态.对金属固体进行缓慢退火就是为了使系统在每一温度下粒子进行充分的热运动而达到平衡态,最终达到稳定的基态.若急剧降温将引起淬火效应,系统只能冷凝为非均匀的亚稳态固体.基于这一原理可以用蒙特卡罗方法模拟退火过程,但必须大量采样才能得到比较精确的结果,因而计算量很大.1953 年米特罗波利斯(Metropolis)以分布函数(1-4)为基础提出了重要性采样的**米特罗波利斯准则**.

第一步,给定以粒子相对位置表征的系统的初始状态 i 作为当前状态,能量为 E_i.

第二步,从状态 i 扰动得状态 j,能量为 E_j,计算能量差 $\Delta E = E_j - E_i$.

第三步,如果 $\Delta E \leqslant 0$,则接受状态 j 为当前状态,即 $j \to i$;否则 $\Delta E > 0$,由分布函数(1-4)来确定:计算分布函数之比

$$r = \frac{\rho_j}{\rho_i} = \exp\left(\frac{-\Delta E}{kT}\right), \tag{1-5}$$

显然有 $0 < r < 1$.选取(0,1)之间的一个随机数 ξ,若 $r > \xi$,则接受状态 j 为当前状态,即 $j \to i$,否则,舍弃 j,当前状态仍为 i.

重复第二、三步,在大量的状态变迁之后,系统处于能量较低的平衡态.降低温度 T 再重复以上过程,系统又处于能量更低的平衡态.

由于 $e^{-x}(x \geqslant 0)$ 是 x 的减函数,从(1-5)式可以看出,温度高时 kT 大,因而 r 较大,接受与当前状态能差较大的新状态的机会多;降低温度,ΔE 大时 r 较小,于是只能接受能差较小的新状态,不断降低温度,最后就只能接受能量较小的新状态.以上模拟退火过程,通过重要性采样的梅特诺波利斯准则,大大减小了计算量.

2 优化中的模拟退火算法

2.1 优化问题

优化问题涉及的范围相当广泛,大体上可分为两类:一类是数值优化(也称函数优化)问题,另一类是组合优化问题.但它们可以描述成统一的形式.

(1) **问题描述** 优化问题用一个偶对 (S,f) 表示,其中 S 表示可行解集,f 是目标函数,它是一个映射:$S \to \mathbf{R}$,求目标函数的最小(大)值问题记为 $\min(\max) f(i), i \in S$.

下面如无特别声明,优化问题均指最小化问题.

(2) **最优解** $i_{opt} \in S$ 称为优化问题 (S,f) 的最优解,如果 $f(i_{opt}) \leqslant f(i)$,对所有 $i \in S$ 成立.

(3) **邻域** 设 $i \in S$，定义某种解的变换方法，若解 $j \in S$ 是由此方法从 i 变换而得，所有这样的 j 的集合称为 i 的一个邻域，j 称为 i 的一个邻近解.

(4) **局部最优解** 设 (S,f) 是一个优化问题，$N \subset S$ 是某个邻域，$i' \in N$. 如果有 $f(i') \leqslant f(i)$ 对所有 $i \in N$ 成立，称 i' 是问题的一个局部最优解.

例 1 流动推销员问题简称 TSP 问题. 给定 n 个城市和每两个城市之间的距离，一个推销员从某个城市出发巡回售货，问这个推销员如何选择路线，使每个城市经过一次且仅一次后回到出发城市，其路径长度最短？

这是一个非常著名的组合优化问题，目前求解这个问题的精确算法都具有呈指数增长的计算复杂性，因此人们转而研究它的近似解法. 其数学描述是：设 $C = [c_{ij}]$ 是距离矩阵，一般地称为价格矩阵，$c_{ij} \geqslant 0$ 是城市 i 到城市 j 的距离(费用)，1, 2, …, n 的一个排列记为 $\pi = (\pi_1 \pi_2 \cdots \pi_n)$，问题是选择怎样的排列 π，使得

$$f(\pi) = \sum_{i=1}^{n} c_{\pi_i \pi_{i+1}}$$

为最小($\pi_{n+1} = \pi_1$).

1, 2, …, n 的一个排列 π 就是 TSP 问题的一个可行解，表示的是一条路径，$\pi_i \pi_{i+1}$ 就是路径中的一条边. 求 TSP 问题的近似解有一种常用的变换称为 k 链变换，也叫 k-opt 变换，其中用得最多的是 2-opt 变换. 如城市数为 10 个，分别编号为 0, 1, …, 9，设它的一个可行解为 $\pi = (2\ 5\ 7\ 8\ 9\ 0\ 1\ 3\ 4\ 6)$. 对它进行 2-opt 变换，是先打断它的两条边，比如说(7,8) 和(3,4)，则回路分成了两个子路径：$4 \to 6 \to 2 \to 5 \to 7$ 与 $8 \to 9 \to 0 \to 1 \to 3$，然后将其中一条路径反向(比如说后者)变为 $3 \to 1 \to 0 \to 9 \to 8$，再与另一条路径相连结，则得到一个新的可行解 $\pi' = (4\ 6\ 2\ 5\ 7\ 3\ 1\ 0\ 9\ 8)$. 用 2-opt 变换求 TSP 问题的局部最优解的计算复杂性是 $O(n^2)$. 类似地可以给出 3-opt 变换，甚至 $(n-1)$-opt 变换，但计算复杂性也不断增加. $(n-1)$-opt 变换实际上已变成了穷举法.

由此例可知，解的变换方法不同，邻域的结构也就不同.

2.2 局部搜索法

设有优化问题 (S,f)，在 S 上定义了某种解的变换方法，局部搜索法如下：

```
Procedure  L_S
begin
   产生初始解 i0;
     i: = i0;
  repeat
     从 Ni 中产生新解 j;
     if(f(j) < f(i)) then i: = j;
  until(f(j) ≥ f(i) 对所有 j ∈ Ni 成立);
end
```

局部搜索法俗称下山法或贪婪算法.

例 2　解0-1背包问题的贪婪法. 一个旅行者要从 n 种物品中选取 b 公斤重的物品,每种物品至多选一件. 问旅行者应怎样挑选,使所选物品的总价值最大?

设第 i 种物品的重量为 W_i,价值为 $c_i, i = 1,2,\cdots,n$,则问题的数学描述为

$$\begin{cases} \max Z = \sum_{i=1}^{n} c_i x_i, \\ \text{s.t.} \sum_{i=1}^{n} W_i x_i \leqslant b, \\ \qquad x_i = \begin{cases} 1, & \text{若第 } i \text{ 种物品被选取}, \\ 0, & \text{否则}. \end{cases} \end{cases}$$

解此问题的贪婪法有三种选取物品的方式,因而也就分别对应于三种解的变换方法.

(1) 使背包总价值增加最快. 将物品按价值从大到小排序,比如 $c_1 \geqslant c_2 \geqslant \cdots \geqslant c_n$. 挑选第 1 到第 k 种物品,使得有 $\sum_{i=1}^{k} W_i \leqslant b < \sum_{i=1}^{k+1} W_i$,于是得近似最优解 $x = (1,\cdots,1,0,\cdots,0)$,其中 x 的前 k 个分量为 1,其余为 0.

(2) 使背包总重量增加最慢,即让背包装下尽量多的物品. 将物品按重量从小到大排序,比如 $W_1 \leqslant W_2 \leqslant \cdots \leqslant W_n$. 余下作法同(1).

(3) 使物品单位重量价值增加最快. 将物品按单位重量价值从大到小排序,比如 $\frac{c_1}{W_1} \geqslant \frac{c_2}{W_2} \geqslant \cdots \geqslant \frac{c_n}{W_n}$. 余下作法同(1).

2.3　从局部搜索法到模拟退火算法

局部搜索法最终必定是落入一个局部最优解的"陷阱"中而不能自拔,能否有一种策略加进算法之中,使之有可能跳出这个陷阱呢?1982 年,Kirkpatrick 从米特罗波利斯模拟金属固体退火过程受到启发,将米特罗波利斯准则引进局部搜索法得到了一种求解优化问题的启发式搜索法,称之为模拟退火算法. 在给出算法描述之前先对热动力学系统与优化问题作一个简单的比较:系统状态对应于可行解;系统能量对应于目标函数;状态转变对应于邻近解;系统温度对应于控制参数;系统基态对应于最优解. 惊人地相似.

基于这样的对比,修改局部搜索法的接受准则为米特罗波利斯准则,得到模拟退火算法.

```
Procedure SAA
begin
  初始化产生初始解和初始控制参数 i0, t0, L0;
  k: = 0;
  i: = i0;
  repeat
```

```
for l: = 0 to L_k do
        begin
          从 N_i 中产生新解 j;
          Δf = f(j) - f(i);
          if(exp(-Δf/t_k) > random(0,1))then i: = j;
        end
   k: = k + 1;
   计算下一个 L_k;
   计算下一个 t_k;
     until 停止准则
end
```

其中，t_k 类似于退火过程中的温度，称为控制参数，必须有 $t_k \to 0$. 最简单的作法是选取 $t_k = \alpha t_{k-1}$，$0 < \alpha < 1$ 称为递减因子，由于需要缓慢退火，α 常取 0.8 左右的值；L_k 为在 t_k 时刻解的变换次数，类似于退火过程中的系统状态转变次数，这里常称为马尔可夫(Markov)链的长度. 注意：当 $f(j) < f(i)$ 时，$\Delta f < 0$，$\exp(-\frac{\Delta f}{t_k}) > 1$，新解 j 一定被接受.

算法的收敛性和计算复杂性主要取决于 t_k(包括 t_0) 和 L_k 的选取，它们也影响停止准则的确定. 还有解的变换方法也是影响算法性能的因素之一.

3　收敛性定理与统计理论

对模拟退火算法进行收敛性分析的理论工具是随机过程理论. 在每一退火控制参数为 t_k 时，算法循环 L_k 次，每次循环相当于一次随机试验，而每次循环的结果仅取决于它前一次循环的结果，因此这是一个马尔可夫链，L_k 也就称为马尔可夫链的长度.

3.1　基本定义

(1) **产生概率**　$\forall\, i、j \in S$，从 i 产生 j 的概率定义为

$$G_{ij}(t_k) = \frac{1}{|N_i|}\varphi_{N_i}(j), \tag{3-1}$$

其中 N_i 是 i 的邻域，$|N_i|$ 表示 N_i 中解的数目. $\varphi_{N_i}(j) = \begin{cases} 1, 当 j \in N_i, \\ 0, 否则, \end{cases}$ 为 N_i 的特征函数. (3-1) 式表明产生概率在邻域 N_i 上是均匀的.

(2) **接受概率**　设 $i \in S$，$j \in N_i$，根据接受准则 j 被接受的概率为

$$A_{ij}(t_k) = \exp\left[-\frac{(f(j) - f(i))^+}{t_k}\right], \tag{3-2}$$

这里，$\forall a \in \mathbf{R}, a^+ = \begin{cases} a, \text{当 } a > 0, \\ 0, \text{否则}. \end{cases}$

(3) **转移概率**　$\forall i、j \in S$，i 到 j 的转移概率定义为

$$P_{ij}(k) = P_{ij}(t_k) = \begin{cases} G_{ij}(t_k)A_{ij}(t_k), & \text{当 } i \neq j, \\ 1 - \sum\limits_{l \in S, l \neq i} P_{il}(t_k), & \text{当 } i = j. \end{cases} \tag{3-3}$$

(4) **以高度 h 可达**　设 $i、j \in S$，称 i 以高度 h 可达 j，若存在 $p \geqslant 1, l_0, l_1, \cdots, l_p \in S$，且 $l_0 = i, l_p = j$，使得 $G_{l_k l_{k+1}} > 0$，且 $f(l_k) \leqslant h$，对所有 $k = 0, 1, \cdots, p-1$ 成立.

(5) **局部极小点的深度**　设 $i^* \in S$ 为一局部极小点，定义 i^* 的深度 $d(i^*)$ 为如下 h 中的最小值：存在 $j \in S$，使 $f(j) < f(i^*)$ 且 i^* 以高度 $f(i^*) + h$ 可达 j. 显然 $d(i_{opt}) = \infty$，i_{opt} 表示最优解.

(6) **平衡分布**　设控制参数 $t_k = t$，转移矩阵为 $P = [p_{ij}(t)]$ 的马尔可夫链的平衡分布定义为向量 $q(t)$，其第 i 个分量 $q_i(t)$ 为 $\forall$ j，

$$q_i(t) = \lim_{k \to \infty} P_r\{X(k) = i \mid X(0) = j\}, \tag{3-4}$$

其中 k 记控制参数为 t 时随机试验的次数，P_r 表示概率，(3-4) 式是一个条件概率，X 是随机变量.

3.2　基本定理

定理 1　设由产生概率形成的产生矩阵 G 满足下列条件：$\forall t > 0, i、j \in S$，存在 $p \geqslant 1, l_0, l_1, \cdots, l_p \in S$，且 $i = l_0, j = l_p$，使得 $G_{l_k l_{k+1}} > 0, k = 0, 1, \cdots, p-1$，则平衡分布 q 存在，其分量为

$$q_i(t) = \frac{|N_i|}{N_0(t)} \exp\left(-\frac{f(i)}{t}\right), \tag{3-5}$$

其中

$$N_0(t) = \sum_{j \in S} |N_j| \exp\left(-\frac{f(j)}{t}\right), \tag{3-6}$$

它类似于统计物理中的系统配分函数. 并且有

$$\lim_{t \to 0} q_i(t) = q_i^* = \frac{|N_i|}{\sum\limits_{j \in S_{opt}} |N_j|} \varphi S_{opt}(i), \tag{3-7}$$

其中 S_{opt} 表示最优解集.

特别地，当 $\forall i \in S, |N_i| = \bar{N}$，即所有解的邻域大小一样时，有

$$\lim_{t \to 0} q_i(t) = \frac{1}{|S_{opt}|} \varphi S_{opt}(i). \tag{3-8}$$

定理 2　设产生矩阵满足条件：

(G_1)　同定理 1 中的条件，

(G_2)　$\forall t > 0, i、j \in S$，有 $G_{ij} = G_{ji}$；

接受矩阵满足条件：

(A_1) $\forall t>0, i、j\in S$，当 $f(i)\geqslant f(j)$ 时有 $A_{ij}(t)=1$，否则，$0<A_{ij}(t)<1$，

(A_2) $\forall t>0, i、j、k\in S$，当 $f(i)\leqslant f(j)\leqslant f(k)$ 时有 $A_{ik}(t)=A_{ij}(t)A_{jk}(t)$，

(A_3) $\forall i、j\in S, f(i)<f(j)$，有 $\lim\limits_{t\to 0}A_{ij}(t)=0$. 则平衡分布 q 存在，其分量为，$\forall i\in S$，有

$$q_i(t)=\frac{A_{i_{\text{opt}}\,i}(t)}{\sum\limits_{j\in S}A_{i_{\text{opt}}\,j}(t)},\tag{3-9}$$

且

$$\lim_{t\to 0}q_i(t)=\frac{1}{|S_{\text{opt}}|}\varphi S_{\text{opt}}(i).\tag{3-10}$$

上述两个定理表明

$$\lim_{t\to 0}\lim_{k\to\infty}P_r(t)\{X(k)\in S_{\text{opt}}\}=1,$$

即模拟退火算法以概率1找到整体最优解集，而收敛到非整体最优解的概率为零.

证明以上两个定理要运用齐次马尔可夫链（即相应的转移概率与试验次数无关）的一些基本性质. 它们虽然保证了算法的收敛性，但要求在每一个控制参数 t_k，L_k 充分大，几乎需要无穷次解的变换才能任意地逼近平衡分布，这显然无法实现. 它们也没有给控制参数 t_k、L_k 等的选取提供任何参考，因为转移概率与 t_k、L_k 无关.

3.3　更精确的收敛性定理

算法实际执行时 L_k 是有限的，稍作改造，模拟退火算法执行的全过程可定义成一个非齐次马尔可夫链（即其转移概率与试验次数相关），从而运用非齐次马尔可夫过程的遍历理论导出更精确的收敛性定理.

设 t'_l 是第 l 个齐次马尔可夫链的控制参数，L'_l 表示该马尔可夫链的长度，$t_k(k=1,2,\cdots)$ 表示第 k 次试验时的控制参数，$\{t_k\}$ 序列定义为

$$t_k=t'_l,\quad lL'_l<k\leqslant(l+1)L'_l,\tag{3-11}$$

即控制参数取成分段常数. 又序列 $\{t'_l\}$，$l=0,1,\cdots$，满足以下条件：

$$t'_{l+1}\leqslant t'_l,\quad l=0,1,\cdots,$$

$$\lim_{l\to\infty}t'_l=0.$$

于是，与试验次数 k 相对应的非齐次马尔可夫链由无限个有限长度的齐次马尔可夫链拼接而成. 这些齐次马尔可夫链的长度可以取得不同，甚至与控制参数相关. 此时的收敛性定理描述 $k\to\infty$ 时非齐次马尔可夫链怎样收敛于(3-8)或(3-10)式的平衡分布.

定理3　设控制参数序列 $\{t_k\}$ 满足

$$t_k\geqslant\frac{(1+r)\Delta}{\ln(k+k_0)},\quad k=0,1,\cdots,\tag{3-12}$$

则与模拟退火算法相应的非齐次马尔可夫链是强遍历的，且收敛到一个最优分布向量，其分量为

$$q_i^* = \frac{1}{|S_{opt}|}\varphi S_{opt}(i). \tag{3-13}$$

其中，$\Delta = \max\limits_{i\in S, j\in N_i}\{|f(i)-f(j)|\}$，$r = \min\limits_{i\in S\setminus S_{max}}\{\max\limits_{j\in S}(d_{ij})\}$，$k_0 \geqslant 2$为一常数．$r$中$S_{max} = \{i\in S \mid f(i)\geqslant f(j), \forall j\in N_i\}$，即局部极大点集，$d_{ij}$为从$i$到$j$所需的最少变换次数，因此$r$为这样的整数，即至少存在一个非局部极大点$i$，从任何其它点$j$到$i$所需的变换数不大于$r$．

必须指出，定理中的常数r也可由一些更大的数取代，如$r = |S|$，即解空间的规模，或者$r = \min\limits_{i\in S\setminus S_{opt}}\{\max\limits_{j\in S}(d_{ij})\}$．定理中产生概率与控制参数$t_k$无关，为了加速算法收敛，产生概率可取其它分布并与$t_k$相关，如利用著名的柯西分布产生新解，有产生概率

$$G_{ij}(t_k) = \frac{t_k}{t_k^2 + d_{ij}^2}. \tag{3-14}$$

定理4 设产生概率由(3-14)式给出，控制参数序列$\{t_k\}$满足(3-12)式，则与模拟退火算法相应的非齐次马尔可夫链是强遍历的，且收敛到一个最优分布向量，其分量如(3-13)式．

类似地，也可以引用其它分布函数，如正态分布等给出产生概率，也有同样的收敛性定理，此处不一一列举．

定理3和定理4中的条件是充分而非必要的．确保渐近收敛的充分且必要条件由Hajek导出，因此下面的定理5也称为Hajek定理．

定理5 模拟退火算法的转移矩阵$P(t_k)$由(3-1)–(3-3)式给出，它收敛于整体最优解，当且仅当

1° $\forall t_k > 0, i、j\in S, \exists p\geqslant 1, l_0, l_1, \cdots, l_p\in S$，使得$l_0 = i, l_p = j$，且$G_{l_k l_{k+1}}(t_k) > 0, k = 0,1,\cdots,p-1$；

2° $\forall i、j\in S, h > 0$，i以高度h可达j，当且仅当j以高度h可达i；

3° 控制参数序列$\{t_k\}$满足

$$t_k \geqslant \frac{D}{\ln(k+2)},\quad k = 0,1,\cdots \tag{3-15}$$

其中

$$D = \max_{i\in S\setminus S_{opt}}\{d(i^*)\} \tag{3-16}$$

即D是局部而非整体极小点的最大深度，$i^*\in N_i$是局部极小点．由于Hajek的条件是充分必要的，因此$D\leqslant\Delta$，Δ由定理3中定义．

3.4 统计特性

设在控制参数为t时，模拟退火算法相应的齐次马尔可夫链收敛到平衡分布$q_i(t)$，其表达式如(3-5)式，则定义若干统计平均量如下．

(1) 目标函数的期望值

$$E_t(f) = \langle f \rangle_t = \sum_{i \in S} f(i) q_i(t). \tag{3-17}$$

(2) 目标函数的平方平均

$$E_t(f^2) = \langle f^2 \rangle_t = \sum_{i \in S} f^2(i) q_i(t). \tag{3-18}$$

(3) 目标函数的均方差

$$\sigma_t^2 = \sum_{i \in S} (f(i) - \langle f \rangle_t)^2 q_i(t). \tag{3-19}$$

(4) 平衡熵

$$s_t = -\sum_{i \in S} q_i(t) \ln q_i(t). \tag{3-20}$$

对于这些统计量,平衡分布为(3-5)式时,有下列基本性质:

1° $\frac{\partial}{\partial t}\langle f \rangle_t = \frac{\sigma_t^2}{t}$.

2° $\frac{\partial}{\partial t} s_t = \frac{\sigma_t^2}{t^3}$.

3° $\lim_{t \to \infty} \langle f \rangle_t = \langle f \rangle_\infty = \frac{1}{|S|} \sum_{i \in S} f(i)$.

4° $\lim_{t \to 0} \langle f \rangle_t = f_{opt}$.

5° $\lim_{t \to \infty} \sigma_t^2 = \sigma_\infty^2 = \frac{1}{|S|} \sum_{i \in S} (f(i) - \langle f \rangle_\infty)^2$.

6° $\lim_{t \to 0} \sigma_t^2 = 0$.

7° $\lim_{t \to \infty} s_t = s_\infty = \ln |S|$.

8° $\lim_{t \to 0} s_t = s_0 = \ln |S_{opt}|$.

性质1到性质8表明,"高温"时"系统"处于无序状态,其熵达到最大值 $\ln |S|$,随着算法执行"温度"不断降低,目标函数期望值和熵分别单调减少至最小值 f_{opt} 和 $\ln |S_{opt}|$,"系统"也达到有序状态.当优化问题的最优解集 $S_{opt} \neq S$ 时,还有下列关于平衡分布 $q_i(t)$ 的性质:

9° $\forall i \in S_{opt}$,有 $\frac{\partial}{\partial t} q_i(t) < 0$.

10° $\forall i \notin S_{opt}, f(i) \geq \langle f \rangle_\infty$,有 $\frac{\partial}{\partial t} q_i(t) > 0$.

11° $\forall i \notin S_{opt}, f(i) < \langle f \rangle_\infty$,则存在 $\tilde{t}_i > 0$,有

$$\frac{\partial}{\partial t} q_i(t) \begin{cases} > 0, & \text{当 } t < \tilde{t}_i, \\ = 0, & \text{当 } t = \tilde{t}_i, \\ < 0, & \text{当 } t > \tilde{t}_i. \end{cases}$$

性质9到性质11给出平衡分布对控制参数的依赖关系.它们表明模拟退火算法找到一个整体最优解的概率随 t 的减小而单调增大;而对于非整体最优解 i 则存

在控制参数的一个确定值 $\tilde{t}_i$,使得 $t < \tilde{t}_i$ 时找到该非最优解的概率随 t 的减小而单调减小.

4　冷却进度表

上节的定理只是从理论上保证了模拟退火算法的渐近收敛性,对算法的实际应用没有多少指导意义.尽管定理4和定理5给出了 t_k 的下界,但其中的 r、h 和 D 等很难估计,实用中并不可行,只能作为参考.而合理选取算法中的控制参数是算法应用的关键,因此人们给出了一些实用的经验法则.

4.1　冷却进度表的一般概念

(1) **冷却进度表**　一个冷却进度表应当规定下述参数:控制参数 t_k 的初始值 t_0;t_k 的衰减函数;t_k 的终值 t_f,即停止准则;马尔可夫链的长度 L_k.

(2) **准平衡**　设控制参数为 t_k 时相应的马尔可夫链的长度为 L_k,称算法达到准平衡,若在 L_k 次变换后,解的概率分布 $\boldsymbol{a}(L_k, t_k)$ 满足:对某些确定的正数 ε 成立

$$\| \boldsymbol{a}(L_k, t_k) - \boldsymbol{q}(t_k) \| < \varepsilon, \tag{4-1}$$

其中 $\boldsymbol{q}(t_k)$ 为由(3-5)式给出的平衡分布.

准平衡的概念是构造冷却进度表的核心,注意(4-1)式只要求对某些 ε 成立,因为对平衡分布任意精度的逼近将导致算法执行时间的指数增长.

(3) **接受率**　模拟退火算法的接受率 χ_k 定义为在给定控制参数 t_k 时,接受的变换数与提出的变换数的比值.

(4) **关于 t_k 与 L_k 的权衡**　设在 t_k 时算法已达到准平衡,若从 t_k 到 t_{k+1} 的衰减量大,对应的平衡分布 $\boldsymbol{q}(t_k)$ 与 $\boldsymbol{q}(t_{k+1})$ 间的差别也大,因而 t_{k+1} 时算法恢复到准平衡需要较长的马尔可夫链.这样,在 t_k 和 L_k 之间存在一种权衡:选取 t_k 较小的衰减以避免过大的 L_k,或者选用较大的 L_k 以使 t_k 能作较大的衰减.

4.2　冷却进度表的确定

确定冷却进度表主要是一些经验法则,下面主要介绍 Aarts 的方法,然后也给出一些其它有代表性的方法.

(1) **控制参数的初值 t_0**　为使算法进程一开始就达到准平衡,应让初始接受率 $\chi_0 \approx 1$.根据米特罗波利斯准则,由 $\exp\left(-\dfrac{\Delta f}{t_0}\right) \approx 1$ 可推知 t_0 值很大.

Kirkpatrick在1982年提出模拟退火算法时给出的经验法则是,选定一个大的 t_0 值,进行若干次变换,如果接受率 χ 小于预定的初始接受率 χ_0(Kirkpatrick 取 $\chi_0 = 0.8$),则将当前 t_0 值加倍,以 t_0 的新值重复上述过程,直到 $\chi > \chi_0$,此时的 t_0 值定为

初始控制参数 t_0.

其后,约翰逊(Johnson)修改这个经验法则不需计算接受率χ,提出经若干次解的变换后,统计目标函数增加的次数,并计算增量Δf的平均值,记为$\overline{\Delta f^+}$,由下式求解 t_0,

$$\chi_0 = \exp\left(-\frac{\overline{\Delta f^+}}{t_0}\right), \tag{4-2}$$

解得

$$t_0 = \frac{\overline{\Delta f^+}}{-\ln\chi_0}. \tag{4-3}$$

Aarts 给出的法则更加精细,先确定初始接受率χ_0,假定对某个控制参数 t_0 产生一个 m_0 次变换的解序列,设 m_1 和 m_2 是这个序列中目标函数减小和增大的变换数,$\overline{\Delta f^+}$ 是 m_2 次变换目标函数增量的平均值,则有下面的近似式:

$$\chi_0 \approx \frac{m_1 + m_2\exp\left(-\frac{\overline{\Delta f^+}}{t_0}\right)}{m_1 + m_2}, \tag{4-4}$$

由此式得

$$t_0 = \frac{\overline{\Delta f^+}}{\ln\dfrac{m_2}{m_2\chi_0 - m_1(1-\chi_0)}}. \tag{4-5}$$

这样的解序列可以产生多个,最终得到的 t_0 作为控制参数初值. 在这里 t_0 的位置实际上已被初始接受率χ_0 取代.

(2) **控制参数的衰减函数**　在求衰减函数时,Aarts 的法则中将准平衡条件(4-1)式代之以

$$\forall k \geqslant 0,\quad \| \boldsymbol{q}(t_k) - \boldsymbol{q}(t_{k+1}) \| < \epsilon \tag{4-6}$$

对某些正数 ϵ 成立.(4-6)式又可化为

$$\forall i \in S,\ \frac{1}{1+\delta} < \frac{q_i(t_k)}{q_i(t_{k+1})} < 1 + \delta,\quad k = 0,1,\cdots, \tag{4-7}$$

其中 δ 与(4-1)式中的 ϵ 相关,是某些小正数.

(4-7)式成立的一个充分条件是

$$\forall i \in S,\ \frac{\exp(-\Delta f_i/t_k)}{\exp(-\Delta f_i/t_{k+1})} < 1 + \delta,\quad k = 0,1,\cdots, \tag{4-8}$$

其中 $\Delta f_i = f(i) - f_{opt}$.(4-8)式可改写为

$$\forall i \in S,\ t_{k+1} > \frac{t_k}{1 + \dfrac{t_k\ln(1+\delta)}{3\sigma_{t_k}}},\quad k = 0,1,\cdots. \tag{4-9}$$

其中 σ_{t_k} 由(3-19)式定义. 将(4-9)式中“>”改成“=”即得衰减函数,式中的参数 δ,Aarts 称它为距离参数.

(3) **控制参数的终值 t_f** Aarts对终值的确定基于目标函数期望值$\langle f\rangle_{t_k}$在$t_k\to 0$时的性质.设

$$\Delta\langle f\rangle_t = \langle f\rangle_t - f_{opt}, \tag{4-10}$$

对某些小正数 ϵ_s,当 $\Delta\langle f\rangle_t < \epsilon_s\langle f\rangle_{t_0}$ 时算法终止.而对于充分大的 t_0 有$\langle f\rangle_{t_0} = \langle f\rangle_\infty$.当 $t \ll 1$ 时,$\Delta\langle f\rangle_t$ 可近似为

$$\Delta\langle f\rangle_t \approx t\,\frac{\partial\langle f\rangle_t}{\partial t}. \tag{4-11}$$

于是,对某些 t_k,若

$$\frac{t_k}{\langle f\rangle_\infty}\cdot\frac{\partial\langle f\rangle_t}{\partial t}\bigg|_{t=t_k} < \epsilon_s \tag{4-12}$$

成立,则算法终止.实际应用中,$\langle f\rangle_{t_k}$用t_k时目标函数的平均值$\bar{f}_{t_k}$近似,或者用 t_k附近若干个相继的$\bar{f}$的平均值近似,以防止过大的波动.

Aarts 称(4-2) 式为**停止准则**,ϵ_s 为**停止参数**.

(4) **关于马尔可夫链的长度 L_k** 由于控制参数的衰减函数由(4-5)式导出,导致控制参数两个相邻值上的平衡分布相互逼近,因此只要在 t_k 值上达到准平衡,则下一个取值 t_{k+1} 上的准平衡只需少量变换就可逼近.这种"少量变换"也要使算法以充分大的概率访问给定解邻域中的大部分解.Aarts证明取 $L_k = \overline{N}$(邻域规模)可以达到上述要求.

Aarts 还证明上述冷却进度表导致马尔可夫链的个数(即算法迭代次数)有上界 $O(\ln|S|)$,而且模拟退火算法的计算时间满足

$$T = O(\tau L\ln|S|). \tag{4-13}$$

其中 τ 表示执行单个变换的计算时间,L 是单个马尔可夫链的长度.对大多数组合优化问题而言,τ 和 L 可选为问题规模的多项式,$\ln|S|$ 也是问题规模的多项式,则此模拟退火算法可在多项式时间里执行.Aarts 也将其冷却进度表称为多项式时间的冷却进度表.

(5) **其它冷却进度表** 控制参数的衰减函数常用的有两种形式:$t_{k+1} = \alpha t_k$ 和 $t_{k+1} = t_k/(1+\beta t_k)$,通过取不同的 α 和 β 值调整衰减速度.Nahar 等人限定控制参数的衰减步数 K,即 K 步算法终止,再把区间$[0, t_0]$ 划分为 K 个小区间,于是衰减函数取为 $t_k = t_0(K-k)/K$, $k = 0,1,\cdots,K-1$.

马尔可夫链的长度通常取为问题规模 n 的多项式或邻域规模 $\overline{N}$ 的多项式,如 $L = 100n$, $L = n$ 或 $L = n^2$ 等,Aarts 的取法就是 $L = \overline{N}$.

停止准则常用的有:限定固定的迭代次数,如 Nahar 的作法;相继 s 个马尔可夫链中解无任何变化则算法终止.Lundy 和 Mees 提出了一个停止准则,是当控制参数 t 满足下式时算法终止:

$$t \leqslant \frac{\epsilon}{[\ln(|S|-1)/\theta]}, \tag{4-14}$$

其中 $|S|$ 是解空间规模,ϵ 和 θ 的意义是得到的解与最优解相差 ϵ 的概率是 θ.还有如接受概率小于某个给定值算法终止等.

5　模拟退火算法的应用

模拟退火算法(参见2.3)主要是重复执行如下过程:产生新解 — 计算目标函数差 — 判断是否接受新解 — 接受或舍弃新解.因此,算法的应用必须解决好三个问题:第一,问题的数学描述;第二,新解的产生和接受机制;第三,冷却进度表.其中第三个问题在3.4节已重点讨论,此处不再赘述.此外,还必须有一个随机数产生器.

问题的数学描述主要包含解空间和目标函数两部分.解空间为问题的所有可行解(有时也包含不可行解)的集合,它限定了初始解选取和新解产生的范围.对无约束优化问题,任一可能解即为一可行解;对有约束优化问题,解集中还包含一些不可行解,对此可以限定解空间仅为可行解的集合,即在构造解时考虑约束条件的限制,也可允许解空间中包含不可行解,而在目标函数中加上罚函数,以惩罚不可行解的出现.目标函数通常表述为若干个优化目标的和式,一般地,目标函数值不一定就是问题待优化的目标值,但其对应关系应是明确的.原则上,目标函数的计算应尽量简单,以免耗时过多.

新解的产生和接受可分为四个步骤:第一,给出解的邻域变换方法,得到一个产生装置,以从当前解产生一个新解;第二,计算新解与当前解的目标函数差,由于目标函数差仅由变换部分产生,因此,通常按增量计算目标函数差;第三,判断新解是否被接受,判断的依据就是米特罗波利斯准则,对于有约束问题往往还伴随有解的可行性判断;第四,接受或舍弃新解,若接受,用新解代替当前解,同时修正目标函数值,若舍弃,保持当前解不变.

下面给出应用模拟退火算法求解组合优化和图论中一些典型NP完全问题的处理方法,以作借鉴,其应用的广泛性可见一斑.需要指出的是,由于模拟退火算法具有鲁棒性(robust),初始解可以随机选取或指定,对最终的求解结果没有影响.

5.1　流动推销员问题

(1) **问题描述**　参见2.1节中的例1.

(2) **解空间**　解空间 S 可表示为 $\{1,\cdots,n\}$ 的所有循环排列的集合,即

$S=\{(\pi_1\pi_2\cdots\pi_n)\mid(\pi_1\pi_2\cdots\pi_n)$ 为 $\{1,\cdots,n\}$ 的循环排列$\}$.其中每一循环排列表示遍访 n 个城市的一条回路,$\pi_i=j$ 表示第 i 次访问城市 j,并约定 $\pi_{n+1}=\pi_1$.初始解可选为$(1\ 2\cdots\ n)$.

(3) **目标函数**　为访问所有城市的总路径长度或总费用,即

$$f(\pi_1,\pi_2,\cdots,\pi_n)=\sum_{i=1}^{n}c_{\pi_i\pi_{i+1}}\quad(\pi_{n+1}=\pi_1).$$

(4) **新解产生**　常用的是2-opt变换或3-opt变换,或若交替地使用这两种变换.

2-opt 变换的实现方法是:随机选取访问的序号 u 和 v(不妨设 $u < v$),逆转 u 和 v 之间的路径顺序,得新解为 $\pi_1\cdots\pi_{u-1}\pi_v\pi_{v-1}\cdots\pi_{u+1}\pi_u\pi_{v+1}\cdots\pi_n$.

3-opt 变换的实现方法是:随机选取访问的序号 u、v 和 w($u \leqslant v < w$),将 u 和 v 之间的路径插到 w 之后,得到新解为 $\pi_1\cdots\pi_{u-1}\pi_{v+1}\cdots\pi_w\pi_u\cdots\pi_v\pi_{w+1}\cdots\pi_n$.

(5) **目标函数差** 对于 2-opt 变换,有目标函数差的计算公式为

$$\Delta f = \Big(c_{\pi_{u-1}\pi_v} + c_{\pi_u\pi_{v+1}} + \sum_{i=u+1}^{v} c_{\pi_i\pi_{i-1}} \Big) - \Big(c_{\pi_{u-1}\pi_u} + c_{\pi_v\pi_{v+1}} + \sum_{i=u+1}^{v} c_{\pi_{i-1}\pi_i} \Big).$$

对于 3-opt 变换有计算公式为

$$\Delta f = \big(c_{\pi_{u-1}\pi_{v+1}} + c_{\pi_w\pi_u} + c_{\pi_v\pi_{w+1}} \big) - \big(c_{\pi_{u-1}\pi_u} + c_{\pi_v\pi_{v+1}} + c_{\pi_w\pi_{w+1}} \big).$$

特别地,当问题为对称时,费用矩阵 $\boldsymbol{C} = [c_{ij}]$ 是对称的. 2-opt变换的计算公式简化为

$$\Delta f = \big(c_{\pi_u\pi_v} + c_{\pi_u\pi_v+1} \big) - \big(c_{\pi_{u-1}\pi_u} + c_{\pi_v\pi_{v+1}} \big).$$

(6) **接受准则** 由于流动推销员问题是最小化问题,接受准则同 2.3 中过程 SAA 的准则.

5.2 最大割问题(MCP 问题)

(1) **问题描述** 给定赋权图 $G = (V,E)$,其中 $V = \{v_1, v_2, \cdots, v_n\}$ 为顶点集,E 为边集,权矩阵为 $\boldsymbol{W} = [w_{ij}]$. 将 V 划分为两个子集 V_0 和 V_1,使边集 E 中那些顶点分属 V_0 和 V_1 的边的权和最大.

(2) **解空间** 可表示为将顶点集 V 划分为两个子集 V_0 与 V_1 的分划 δ 的集合,即

$$S = \{\delta \mid \delta \text{ 为将 } V \text{ 划分为 } V_0 \text{ 与 } V_1 \text{ 的一个分划}\}.$$

其中分划

$$\delta(i) = j, \quad i = 1, \cdots, n, j = 0, 1,$$

表示顶点 $v_i \in V_j$. 初始解可选 $\delta(i) = 0, i = 1, \cdots, n$.

(3) **目标函数** 取分划 δ 所得到的割量(cut)

$$f(\delta) = \sum_{\delta(i) \neq \delta(k)} w_{ik},$$

需求其最大值.

(4) **新解产生** 任选顶点 $v_k \in V$,将其从当前所在子集移到另一个子集,即

$$\delta(k) = 1 - \delta(k).$$

(5) **目标函数差** 顶点 v_k 移动后导致目标函数有增量,从当前解求得为

$$\Delta f = \sum_{\delta(i) = \delta(k)} w_{ik} - \sum_{\delta(i) \neq \delta(k)} w_{ik}.$$

(6) **接受准则** MCP 问题为一最大化问题,接受概率应为

$$P_r = \begin{cases} 1, & \Delta f > 0, \\ \exp\left(\dfrac{\Delta f}{t}\right), & \text{否则}. \end{cases}$$

5.3　0-1 背包问题(ZKP 问题)

(1) **问题描述**　给定一个可装重量 M 的背包及 n 件物品,物品 i 的重量和价值分别为 W_i 与 $c_i(i = 1,\cdots,n)$.要选若干件物品装入背包,使其价值之和最大.

(2) **解空间**　ZKP 问题是一个有约束的优化问题,故限定解空间为可行解的集合,即

$$S = \{(x_1,\cdots,x_n) \mid \sum_{i=1}^{n} W_i x_i \leqslant M, \quad x_i = 0 \text{或} 1\},$$

其中 $x_i = 1$ 表示物品 i 被选入背包.初始解选为(0,…,0).

(3) **目标函数**　在解空间中求下列函数的最大值,

$$f(x_1,\cdots,x_n) = \sum_{i=1}^{n} c_i x_i .$$

(4) **新解产生**　随机地选取物品 i,若 i 不在背包中,则将其装入背包中或者装入 i 并取出 j;若 i 在背包中,则将其取出并装入 j.即有三种产生方式:$x_i = 1 - x_i$;$x_i = 1 - x_i$ 且 $x_j = 1 - x_j(i \neq j)$.

(5) **目标函数差**　根据产生新解的三种可能方式,目标函数差即背包价值差为

$$\Delta f = \begin{cases} c_i, & \text{装入物品 } i, \\ c_i - c_j, & \text{装入 } i \text{ 取出 } j, \\ c_j - c_i, & \text{取出 } i \text{ 装入 } j. \end{cases}$$

为判定解的可行性,还需有对应的背包重量差,即背包重量的增量

$$\Delta m = \begin{cases} W_i, & \text{装入物品 } i, \\ W_i - W_j, & \text{装入 } i \text{ 取出 } j, \\ W_j - W_i, & \text{取出 } i \text{ 装入 } j. \end{cases}$$

(6) **接受准则**　加入解的可行性判断的接受概率

$$P_r = \begin{cases} 0, & m + \Delta m > M \\ 1, & m + \Delta m \leqslant M \text{ 且 } \Delta f > 0, \\ \exp\left(\dfrac{\Delta f}{t}\right), & \text{其它}. \end{cases}$$

5.4　独立集问题(ISP 问题)

(1) **问题描述**　设有图 $G = (V,E)$,$V = \{v_1,\cdots,v_n\}$ 为顶点集,E 为边集.要求 V 的最大独立集 V',满足 $\forall u、v \in V'$ 有 $\{u,v\} \overline{\in} E$.

(2) **解空间**　取为 V 的幂集 2^V,即 V 的所有子集 V' 的集合:

$$S = \{V' \mid V' \subseteq V\}.$$

注意解空间 S 中可能含有不可行解,即含有非独立的顶点子集.初始解取 $V' = \varnothing$.

适当拓宽解空间可以使产生新解的方法简便，同时使算法增大逃离局部最优陷阱的概率.当然，目标函数的构造会复杂些.

(3) **目标函数**　为惩罚可能出现的不可行解，目标函数选为

$$f(V')=|V'|-\lambda|E'|,$$

其中 $|V'|$ 表示集 V' 中的顶点数，$E'\subseteq E$ 为

$$E'=\{\{u,v\}\mid u、v\in V',\{u,v\}\in E\},$$

$\lambda>1$，是一个罚函数因子，用于惩罚在 V' 中存在的边.当 $E'=\varnothing$ 时，V' 为独立集，$f(V')=|V'|$ 恰好是 V' 中的顶点数.

(4) **新解产生**　随机地在 V 中选取顶点 u，若 $u\in V'$，则将 u 移出 V'，若 $u\bar{\in}V'$，将 u 移入 V'，得到新解 V''.用集 V' 的特征函数表示就是

$$\varphi_{V''}(u)=1-\varphi_{V'}(u).$$

(5) **目标函数差**　设 $A=[a_{ij}]$ 表示图 G 的邻接矩阵，$a_{ij}=1$ 表示顶点 v_i 与 v_j 邻接，即 $\{v_i,v_j\}\in E$.用集 V' 的特征函数和邻接矩阵将目标函数具体写出就是

$$f(V')=\sum_{u\in V}\varphi_{V'}(u)-\lambda\sum_{u,v\in V'}a_{uv}.$$

当选定 $u\in V$ 产生新解时，有目标函数差

$$\Delta f=\begin{cases}-\left(1-\lambda\sum\limits_{v\in V'}a_{uv}\right), & u\in V',\\ 1-\lambda\sum\limits_{v\in V'}a_{uv}, & u\bar{\in}V'.\end{cases}$$

运用特征函数写成紧凑形式有

$$\Delta f=(\varphi_{V\setminus V'}(u)-\varphi_{V'}(u))\left(1-\lambda\sum_{v\in V'}a_{uv}\right).$$

(6) **接受准则**　ISP 问题为最大化问题，与 MCP 问题一样，接受概率为

$$P_r=\begin{cases}1, & \Delta f>0,\\ \exp\left(\dfrac{\Delta f}{t}\right), & \text{否则}.\end{cases}$$

5.5　调度问题(SCP 问题)

(1) **问题描述**　有 n 个相互独立的任务 $J_1,J_2,\cdots,J_n$，所需时间分别为 $t_1,t_2,\cdots,t_n$，均可由 m 台机器 $M_1,M_2,\cdots,M_m$ 中的任一台完成，但每台机器一次仅可完成一个任务.要寻找一个最小调度，即找一个将 n 个任务分配给 m 台机器的调度，使完成所有任务的时间最短.

(2) **解空间**　一个调度可看成对正数集 $T=\{t_1,t_2,\cdots,t_n\}$ 的一个分划 $\{T_1,T_2,\cdots,T_m\}$，$\bigcup\limits_{i=1}^{m}T_i=T$，$T_i\cap T_j=\varnothing$，$i\neq j$.因此，解空间便由所有可能的分划组成，即

$$S=\left\{\{T_1,\cdots,T_m\}\mid\bigcup_{i=1}^{m}T_i=T,T_i\cap T_j=\varnothing,i\neq j\right\}.$$

初始解选为$\{T,\varnothing,\cdots,\varnothing\}$.

(3) **目标函数** 求解SCP问题就是要使分划$\{T_1,T_2,\cdots,T_m\}$的各子集T_i内的正数之和的最大值为最小,即求

$$\max_{1\leq i\leq m}\{\sum_{t\in T_i}t\}$$

的最小值.易证求其最小值对应于求

$$\sum_{i=1}^{m}(\sum_{t\in T_i}t)^2$$

的最小值,所以目标函数可定义为

$$f(T_1,T_2,\cdots,T_m)=\sum_{i=1}^{m}(\sum_{t\in T_i}t)^2.$$

注意此时求出的f的最小值并非实际调度的时间.

(4) **新解产生** 任选$t\in T_i$,将其移到任选的另一子集T_j中去,即

$$T_i=T_i\setminus\{t\},\quad T_j=T_j\cup\{t\},\quad i\neq j.$$

它对应于将所需时间为t的某个任务从机器M_i调到M_j上去完成.

(5) **目标函数差** 根据新解产生的方式,从当前解求目标函数差为

$$\Delta f=2t(\sum_{t'\in T_j}t'-\sum_{t''\in T_i}t''+t).$$

注意在当前解中,$t\in T_i$,第二个和式中的$-2t^2$正好与$2t^2$相抵销.

(6) **接受准则** 最小化问题保持过程SAA中的接受准则不变.

注意SCP问题的目标函数有一个确定的且有时可以达到的下界$(\sum_{i=1}^{n}t_i)^2/m$,它对应于所有m台机器所需时间均为$(\sum_{i=1}^{n}t_i)/m$,因此可在停止准则中增加一个"if $f=(\sum_{i=1}^{n}t_i)^2/m$ *then* 算法终止"的判断,这样得到的调度必为一个最小调度.

5.6 划分问题(PAP问题)

(1) **问题描述** 设有正数集$T=\{t_1,t_2,\cdots,t_n\}$,要找集T的一个最接近分划,即将T划分为子集T_0和T_1的分划,使两个子集中的正数之和最接近.

不考虑问题的具体意义,抽象地看,PAP问题是上述SCP问题当$m=2$时的特例,因此上述解SCP的算法描述完全适用于PAP问题,但PAP问题也有其特殊性.

(2) **解空间** 所有将T划分为子集T_0和T_1的分划的集合,即

$$S=\{\{T_0,T_1\}\mid T_0\cup T_1=T,T_0\cap T_1=\varnothing\}.$$

初始解选为$\{T,\varnothing\}$.

(3) **目标函数** 定义为两个子集中各正数之和的差的绝对值,即

$$f(T_0,T_1)=\left|\sum_{t\in T_0}t-\sum_{t\in T_1}t\right|.$$

显然应求最小值.

(4) **新解产生** 任选正数 $t\in T_i(i=0,1)$,将其移到子集 T_{1-i} 中去.若用 $\varphi(t)=i$ 表示 $t\in T_i$,则有

$$\varphi(t)=1-\varphi(t).$$

(5) **计算目标函数差** 记当前解 $\{T_0,T_1\}$ 中和较大的子集为 T_j,则由当前解计算目标函数差为

$$\Delta f=\begin{cases}2t, & \varphi(t)\neq j,\\ -2t, & \varphi(t)=j\text{ 且 }f\geqslant 2t,\\ 2t-2f, & \varphi(t)=j\text{ 且 }f<2t.\end{cases}$$

同时 j 也需作相应的变化

$$j=\begin{cases}j, & \varphi(t)\neq j,\\ j, & \varphi(t)=j\text{ 且 }f\geqslant 2t,\\ 1-j, & \varphi(t)=j\text{ 且 }f<2t.\end{cases}$$

(6) **接受准则** 保持过程 SAA 中准则不变.

与 SCP 问题类似,停止准则增加一个"if $f=0$ then 算法终止"的判断.

5.7 布局问题(PLP 问题)

(1) **问题描述** 设有 n 个矩形块 $B_1,B_2,\cdots,B_n$,各 B_i 和 B_j 之间的权为 $w_{ij}(i,j=1,\cdots,n)$.要找一个最优布局,将 $B_1,B_2,\cdots,B_n$ 放置到一个包络矩形内,使各 B_i 之间没有重叠,且包络矩形的面积 A 与和式

$$c=\sum_{i=1}^{n}\sum_{j=i+1}^{n}d_{ij}w_{ij}$$

之和 $A+\lambda_\omega c$ 为最小,其中 d_{ij} 表示布局中矩形块 B_i 和 B_j 之间的距离,λ_ω 表示 c 在和式 $A+\lambda_\omega c$ 中的相对权的正权因子.

布局问题有很广泛的实际应用背景,例如超大规模集成电路(VLSI)的设计,此时 w_{ij} 相应于集成块 B_i 和 B_j 之间的连通特性(connectivity),又如设备布置,此时 w_{ij} 表示设备 B_i 和 B_j 之间的毗邻要求.

(2) **解空间** 取 S 为对矩形块 $B_1,B_2,\cdots,B_n$ 的所有布局的集合,即

$$S=\{P\mid P\text{ 是 }n\text{ 个矩形块的一个布局}\},$$

其中 P 可能是不可行解,即有重叠的布局.初始解可选为任一随机布局.

(3) **目标函数** 定义为一个需求最小值的和式,即

$$f(P)=A+\lambda_\omega c+\lambda_0 f_0,$$

其中 A、λ_ω 和 c 的含义与(1)中相同,f_0 为布局 P 中矩形块重叠的数目,λ_0 是一个罚函数因子,用于惩罚 P 中出现的重叠.λ_0 的值应取得很大.如何取为 λ'_0/t,这样随着控制参数 t 的递减并趋于零,重叠可完全排除.

(4) **新解产生** 通过调整当前布局来产生,通常选取对单个矩形块的重置,如

平移、旋转、反转或两个矩形块互换等.例如,选矩形块 B_i 移到另一处产生新解.

(5) **目标函数差**　与新解产生方法有关.如对(4)中的例子有

$$\Delta f = \Delta A + \lambda_w \sum_{j=1}^{n} \Delta d_{ji} w_{ji} + \lambda_0 \Delta f_0,$$

其中 ΔA 为包络矩形面积的增量,Δd_{ji} 为 B_i 与 B_j 之间距离的增量,Δf_0 为矩形块重叠数的增量.

(6) **接受准则**　保持过程 SAA 中的接受准则不变.

5.8　图的着色问题(GCP问题)

(1) **问题描述**　给定图 $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$ 为顶点集,E 为边集.要寻找图 G 的一个最小着色,即找一个最小的正整数 k 和映射 $\varphi: V \to \{1, \cdots, k\}$,使得 $\forall\, u、v \in V$,当 $\{u, v\} \in E$ 时有 $\varphi(u) \neq \varphi(v)$.

图的着色问题可以看成是将顶点集 V 划分成最少个数独立集的问题,集 V 的一个分划(独立集个数不一定最少)就是一种着色.若图 G 的度数(即 G 中顶点度数的最大值)为 d,则图 G 的着色有一个上界为 $d+1$.引入一个正权集合 $W = \{w_1, w_2, \cdots, w_{d+1}\}$,颜色 i 被赋权值 w_i,下面的等价性定理将寻找最小着色转化成为求一个目标函数的最大值.

定理6　设 W 中各权 $w_j (j = 1, \cdots, d)$ 之间满足下列递推关系:

$$w_{j+1} < \left(\frac{|V|}{j} - 1\right) \sum_{i=2}^{j} w_i - \left(\frac{|V|(j-1)}{j} - j\right) w_1,$$

则图的着色问题等价于找 V 的分划 $\delta = \{V_1, V_2, \cdots, V_{d+1}\}$,使得 $\forall\, v_i、v_j \in V_k$, $\{v_i, v_j\} \bar{\in} E$,且

$$f(\delta) = \sum_{i=1}^{d+1} w_i |V_i|$$

最大.

实际求解时 w_j 之间的递推关系可代之以

$$w_{j+1} < 2w_j - w_1, \quad j = 1, \cdots, d.$$

但在此条件下使 $f(\delta)$ 最大的分划 δ 不一定对应图 G 的最小着色.

(2) **解空间**　记将顶点集 V 划分为 $d+1$ 个子集的分划为 δ,即

$$\delta = \{V_1, V_2, \cdots, V_{d+1}\}, \quad \bigcup_{i=1}^{d+1} V_i = V, \quad V_i \cap V_j = \varnothing, i \neq j,$$

则解空间表示为

$$S = \{\delta \mid \delta \text{ 是 } V \text{ 的一个分划}\}.$$

注意有些分划可能并非可行解,初始解取为 $\delta = \{V, \varnothing, \cdots, \varnothing\}$.

(3) **目标函数**　为了惩罚不可行的分划,目标函数中加进了惩罚项,取为

$$f(\delta) = \sum_{i=1}^{d+1} w_i (|V_i| - \lambda |E_i|),$$

其中 $$E_i = \{\{u,v\} \mid u,v \in V_i, \{u,v\} \in E\}$$
为 V_i 中的边的集合，λ 为一大于 1 的罚函数因子.

(4) **新解产生**　任选顶点 $u \in V$，将其从当前所在子集 V_i 移到任选的另一子集 V_j 中去，即

$$V_i = V_i \setminus \{u\}, \quad V_j = V_j \cup \{u\}, i \neq j.$$

(5) **目标函数差**　设 $A = [a_{ij}]$ 为图 G 的邻接矩阵，$a_{ij} = 1$ 当且仅当 $\{v_i, v_j\} \in E$，则有目标函数差为

$$\Delta f = w_j - w_i - \lambda\Big(w_j \sum_{v \in V_j} a_{uv} - w_i \sum_{v \in V_i} a_{uv}\Big).$$

(6) **接受准则**　因图的着色问题是最大化问题，接受概率为

$$P_r = \begin{cases} 1, & \Delta f > 0, \\ \exp\left(\dfrac{\Delta f}{t}\right), & \text{否则}. \end{cases}$$

参 考 文 献

1 康立山，谢云，尤矢勇，罗祖华著．非数值并行算法(第一册)——模拟退火算法．北京：科学出版社，1994．

2 Aarts E, Korst J. Simulated annealing and Boltzmann machine. New York: John Wiley and Sons, 1989.

3 Colin R Reeves. Modern heuristic techniques for combinatorial problems. London: Blackwell Scientific Publications, 1993.

·计算机数学卷·

第 16 篇

数学机械化与机械化数学

编　者　高小山　林东岱　石　赫
审校者　吴文达

目　录

引　言

17 世纪以来，人类历史上经历了一场史无前例的技术革命，出现了以蒸汽机为代表的机器，代替各种类型的体力劳动，使人类进入了一个新时代．如果说工业机器的出现所导致的产业革命使人们逐渐实现了体力劳动的机械化，从而促进了社会生产力的发展，那么 20 世纪电子计算机的产生，则为人类实现脑力劳动的机械化创造了物质条件．当前在一些领域里，计算机已经开始能部分地代替人类的脑力劳动．数学是各类学科的基础，具有表达精确、论证严谨等特点，是一个典型的脑力劳动．数学问题的机械化，无疑在脑力劳动机械化的过程中将起着重要的作用．

所谓机械化，无非就是刻板化与规格化．无论是机器代替人的体力劳动（如纺纱织布），还是计算机代替某种形式的脑力劳动（如加减乘除），之所以成为可能，关键在于所代替的劳动已经实现了规格化与刻板化．因此数学问题的机械化，就是要将数学问题刻板化与规格化，也就是说在计算或证明过程中，每前进一步都应有一个确定的必须选择的下一步，从而可以按照一条有规律的刻板的道路，一直到达结论．

历史上每次工具的革新，都会某种形式、某种程度地促进生产力的发展与技术的进步．同样，数学研究上工具的变革，无疑也会对数学科学的发展产生一定的影响．在今天，电子计算机飞速发展，技术不断提高，数学中人们难以胜任的繁琐而重复的大量计算已经可用计算机代替，数学机械化的思想无疑有着广阔的发展前景．

回顾数学发展的历史，主要有两种中心思想：一是公理化思想，另一是机械化思想．前者源于希腊，后者则贯穿整个中国古代数学．从汉初完成的《九章算术》中对开平方、开立方的机械化过程的详细描述到宋元时代发展起来的高次代数方程的数值解法，无一不与数学机械化思想有关，并对数学的发展起了具大的作用．但由于各种原因，在现代数学中，特别是纯粹数学中，机械化的思想却没有得到应有的重视．

20 世纪 70 年代，我国数学家吴文俊先生从中国古典数学思想出发，从几何定理证明入手所建立的一套数学机械化方法，不仅将中国传统数学发扬光大，而且也为国际自动推理的研究开辟了新的前景，而被国际学术界誉为“吴方法”．经过近二十年的努力，几何定理自动证明的吴方法及在其影响下产生的一系列重要的新方法，已经发展成具有我国特色的数学机械化理论．这一理论已不只在几何定理的机器证明、方程组求解、微分几何、理论物理等领域得到成功应用，还为机器人学、几何辅助设计、CAD、计算机视觉等高科技领域提供了有力工具．本篇将简要介绍机械化数学的主要内容．

1 构造性代数几何

代数几何是近代数学中十分活跃的分支.长期以来,形成了各种流派.当前流行的代数几何其研究方法大都是存在性的.机械化数学强调数学的构造性研究,而构造性代数几何理论则是机械化数学的重要理论基础.本章介绍以吴-Ritt 分解算法为核心的构造性代数几何的主要内容.

1.1 吴-Ritt 零点分解算法

设 K 为一特征为 0 的域, $K[x_1,x_2,\cdots,x_N]$ 是 K 上以 $x_1,x_2,\cdots,x_N$ 为变元的所有多项式组成的多项式环, $K(x_1,x_2,\cdots,x_N)$ 是 $K[x_1,x_2,\cdots,x_N]$ 的分式域.对任意多项式 $f\in K[x_1,x_2,\cdots,x_N]$,用 $\deg_{x_i}(f)$ 表示变元 x_i 在多项式 f 中实际出现的最高次数.不出现任何变元的多项式称为常数多项式.

假设变元 $x_1,x_2,\cdots,x_N$ 有次序关系 $x_1<x_2<\cdots<x_N$.对 $K[x_1,x_2,\cdots,x_N]$ 中任一多项式 f, f 的类 $\mathrm{cls}(f)$ 是一非负整数,定义为 f 中实际出现的变元的最大下标.如果 f 为常数多项式,则定义为 0.假设 $f\in K[x_1,x_2,\cdots,x_N]$, $\mathrm{cls}(f)=m\neq 0$,则 f 可以写为

$$f = Ix_m^d + x_m \text{ 的低次项}.$$

其中 $d=\deg_{x_m}(f)$, $I\in K[x_1,x_2,\cdots,x_{m-1}]$.一般来说 I 是 $x_1,x_2,\cdots,x_{m-1}$ 的多项式,称 I 为多项式 f 的**初式**,记作 $\mathrm{Init}(f)$. x_m 有时也被称为多项式 f 的**主变元**.

设 $f\in K[x_1,x_2,\cdots,x_N]$, $\mathrm{cls}(f)=m>0$, $K[x_1,x_2,\cdots,x_N]$ 中任一多项式 g 称为对 f **已约化的多项式**,如果 $\deg_{x_m}(g)<\deg_{x_m}(f)$.显然多项式 f 的初式对 f 已约化.

定义 1(伪除法) 设 $f\in K[x_1,x_2,\cdots,x_N]$ 为一非常数多项式,初式为 I,则对任一多项式 $g\in K[x_1,x_2,\cdots,x_N]$,总存在非负整数 s 及多项式 Q、$R\in K[x_1,x_2,\cdots,x_N]$,使得 $I^sg=Qf+R$ 且 R 对 f 已约化.当 s 为能有这种形式的最小非负整数时,称 R 为 g 对 f 的余式,记作 $\mathrm{prem}(g,f)$.

$R=\mathrm{prem}(g,f)$ 的计算过程如下:

设 $p=\mathrm{cls}(f)$, $I=\mathrm{Init}(f)$, $d_1=\deg_{x_p}(f)$.若 $p=0$ 则 R 为零,否则令 $R=g$.重复下面过程直到 R 对 f 约化:设 $d_2=\deg_{x_p}(r)$, $c=R$ 作为 x_p 的多项式的首项系数, $R=IR-x_p^{d_2-d_1}f$.

显然任一非零多项式对其自身的余式为 0.以后约定任一多项式对一非零常数多项式的余式总为 0.

定义 2 设 $A:f_1,f_2,\cdots,f_r$ 为一多项式的有限序列.称 A 为**一升列**,如果

1° $r=1$ 但 $f_1\neq 0$;

或 2° $r>0$, $0<\mathrm{cls}(f_1)<\mathrm{cls}(f_2)<\cdots<\mathrm{cls}(f_r)$,且对任意的 $j>i$, f_j 对 f_i 已约

化.

显然总有升列的长度 $r \leq N$. 有时把单一非零常数多项式组成的升列称为矛盾列.

设 $f_1, f_2, \cdots, f_r$ 是一升列, 任一多项式 g 对该升列的余式 $\mathrm{prem}(g, f_1, \cdots, f_r)$ 归纳地定义为 $\mathrm{prem}(\mathrm{prem}(g, f_r), f_1, \cdots, f_{r-1})$, 令其为 R, 则有余式公式:

$$I_1^{s_1} I_2^{s_2} \cdots I_r^{s_r} g = Q_1 f_1 + \cdots + Q_r f_r + R,$$

其中 I_i 是 f_i 的初式, s_i 为非负整数, $Q_1, Q_2, \cdots, Q_r$ 都是 $K[x_1, x_2, \cdots, x_N]$ 中的多项式且 R 对诸 f_i 都已约化.

有了多项式升列的概念, 可进一步引入多项式组的特征列. 先看下面的定义.

定义 3　设 f 和 g 是 $K[x_1, x_2, \cdots, x_n]$ 中的两个非 0 多项式, 称 f 比 g 有较低的秩或 g 比 f 有较高的秩, 记作 $f < g$ 或 $g > f$, 如果

1° $\mathrm{cls}(f) < \mathrm{cls}(g)$

或　2° $\mathrm{cls}(f) = \mathrm{cls}(g) = p \neq 0$, 且 $\deg_{x_p} f < \deg_{x_p} g$.

当两多项式 f 和 g 不能比较秩的高低时, 称 f 和 g 有相同的秩, 记作 $f \sim g$.

显然对任一多项式的非空集合, 按秩的高低, 总存在一极小元素, 即秩不高于其它任何多项式的多项式.

定义 4　设 $A: f_1, f_2, \cdots, f_r; B: g_1, g_2, \cdots, g_s$ 为两多项式列, 称 A 比 B 有较高的秩或 B 比 A 有较低的秩, 记作 $A > B$ 或 $B < A$, 如果

1° 存在 $j \leq \min(r, s)$, 使得 $g_1 \sim f_1, \cdots, f_{j-1} \sim g_{j-1}, f_j > g_j$;

或　2° $s > r$ 且 $f_1 \sim g_1, \cdots, f_r \sim g_r$.

在两个升列 A 和 B 不能比较秩的高低时, 称其有相同的秩, 记作 $A \sim B$.

显然升列对秩的高低来说构成一部分次序, 因此, 对一个升列的集合, 可以引入极小升列的概念, 即集合中秩不高于其它任何升列的升列.

引理 1　设 $A_1, A_2, \cdots, A_q, \cdots$ 是一升列的非增秩序列, 则一定存在 q 使得 $A_q, A_{q+1}, \cdots$ 都具有相同的秩.

从上述定理出发有:

推论 1　对任一升列的非空集合, 一定有极小升列存在.

一个升列 A 称属于多项式集合 PS, 如果 A 中每一多项式都是 PS 中的多项式. 由于任一非零多项式都单独构成一升列, 因此属于非空多项式集合 PS 的升列一定存在, 故属于 PS 的升列构成一非空集合. 由推论 1 从而一定有极小升列. 任一这样的极小升列, 都称为 PS 的一个**基列**.

综上所述有,

引理 2　设 PS 为一多项式的非空集合, 则 PS 必有基列存在.

下面是求多项式集合 PS 的基列 AS 的一个算法.

首先设 AS 为空集, PS′ = PS. P 是 PS′ 中秩最低的多项式, AS = AS ∪ {P}, PS′ = PS′ 中对 AS 已约化的所有多项式. 然后对 PS′ 重复上述过程直至 PS′ 变为空集合为止.

对于多项式集合的基列, 有如下引理:

引理3 设PS为一多项式的非空集合,A为其一基列.假设g是一对A已约化的非0多项式,则PS $\cup \{g\}$ 的基列比A必有较低的秩.

设PS为一多项式的非空集合,结合引理1和引理3考察如下步骤:

$$\begin{array}{llll} PS_1 = PS, & PS_2 = PS \cup RS_1, & \cdots, & PS_n = PS \cup RS_{n-1}, \\ BS_1, & BS_2, & \cdots, & BS_n, \cdots \\ RS_1, & RS_2 & \cdots, & RS_n, \cdots \end{array}$$

其中BS_i为PS_i的基列,RS_i是PS_i中多项式对BS_i的所有非0余式组成的集合.根据引理3可知$BS_1, BS_2, \cdots$是一升列的非增秩序列,故一定存在一正整数n'使得$BS_{n'} \sim BS_{n'+1} \sim BS_{n'+2} \sim \cdots$,这时将有$RS_{n'} = RS_{n'+1} = \cdots = \varnothing$,也就是说PS中的任一多项式都将被$BS_{n'}$约化成0.

综上所述有下面整序原理:

定理1 设PS为一多项式的非空集合,有一机械化算法可在有限步内求得一升列CS:$f_1, f_2, \cdots, f_r$,使得$f_i \in$ Ideal(PS)(PS生成的理想),且对任一多项式$f \in$ PS, prem$(f, CS) = 0$.

设E是K的一个扩域,用Zero(S)表示S中的多项式在E上的公共零点组成的集合,即

$$\mathrm{Zero}(S) = \{(a_1, a_2, \cdots, a_N) \in E^N \mid h(a_1, a_2, \cdots, a_N) = 0, \text{对所有 } h \in S\}.$$

假设G是另一多项式集合,Zero(S/G)表示是S的但不是G中所有多项式的零点所组成的集合.对于单一多项式Q,Zero($S/\{Q\}$)有时也写为Zero(S/Q).

定理2(整序原理) 设PS为一多项式的非空集合,CS是PS的一特征列,则

$$\mathrm{Zero}(CS/J) \subset \mathrm{Zero}(PS) \subset \mathrm{Zero}(CS)$$

$$\mathrm{Zero}(PS) = \mathrm{Zero}(CS/J) \cup \mathrm{Zero}(\bigcup_i PS \cup \{I_i\})$$

其中I_i是CS中多项式的初式,J为所有I_i之乘积.

显然当所求得的特征列CS为一矛盾列时,有Zero(PS) = Zero(CS) = $\varnothing$.这说明PS中的多项式没有公共的零点.在上述定理中,如果对诸Zero(PS $\cup \{I_i\}$)继续进行分解,可得

定理3 (零点分解定理 —— 弱形式) 对一非空多项式集合PS与任一多项式G,有分解

$$\mathrm{Zero}(PS/G) = \bigcup_k \mathrm{Zero}(ASC_k/J_k G_k)$$

其中ASC_k是升列,J_k为ASC_k中多项式的初式之乘积,G_k是G对ASC_k的余式且$G_k \neq 0$.(参阅文献[1])

设ASC:$f_1, f_2, \cdots, f_r$为一升列,$c_i = \mathrm{cls}(f_i)$.称ASC是一不可约升列,如果f_1在$K_1 = K[x_1, x_2, \cdots, x_{c-1}]$上作为$x_{c_1}$的多项式是不可约的;并且如果$f_i$在$K_i$上作为$x_c$的多项式是不可约的,添加$f_i$在某一扩域的根$\alpha_i$到$K_i$得到扩域$K_{i+1} = K_i(\alpha_i)$,则$f_{i+1}$在$K_{i+1}$上作为$x_{c_{i+1}}$的多项式也是不可约的.

考察定理2中分解,通过对诸ASC_k中的多项式依次做代数扩域上的因式分解,可得

定理 4(零点分解定理——强形式,[1]) 对任意多项式的非空集合 PS 和多项式集合 G,存在一算法,在有限步内或者判定 Zero(PS/G) 为空集或者找出一组不可约升列 IRR_i 满足

$$\mathrm{Zero}(\mathrm{PS}/G) = \bigcup_i \mathrm{Zero}(\mathrm{IRR}_i/\{I_i\} \cup G),$$

其中 $\{I_i\}$ 是 IRR_i 中多项式的初式构成的集合,并且对任意 i 和 $g \in G$, $\mathrm{prem}(g, \mathrm{IRR}_i) \neq 0$.

设 ASC 为一升列,定义

$$\mathrm{PD}(\mathrm{ASC}) = \{P \mid \mathrm{prem}(P, \mathrm{ASC}) = 0\},$$

$$\mathrm{QD}(\mathrm{ASC}) = \{P \mid \text{存在 ASC 初式之积 } J \text{ 使得 } JP \in \mathrm{Ideal}(\mathrm{ASC})\}.$$

定理 5 有

1° $\mathrm{PD}(\mathrm{ASC}) \subsetneq \mathrm{QD}(\mathrm{ASC})$.

2° 对于不可约升列 ASC, PD(ASC) = QD(ASC) 是一个素理想.

3° 对于任意升列 ASC, QD(ASC) 或为 $K[x]$ 或是一个单纯理想,即 QD(ASC) 的所有非多余素理想分支均有相同的维数(见文献[2]).

利用这一符号,可以得到下面形式的零点分解定理(详见文献[1]).

定理 6(代数簇的分解) 对任意多项式的非空集合 PS 和多项式 G,如果存在下面形式的分解

$$\mathrm{Zero}(\mathrm{PS}/G) = \bigcup_i \mathrm{Zero}(\mathrm{ASC}_i/\{I_i\} \cup G),$$

则有

$$\mathrm{Zero}(\mathrm{PS}/G) = \bigcup_i \mathrm{Zero}(\mathrm{PD}(\mathrm{ASC}_i)/G),$$

$$\mathrm{Zero}(\mathrm{PS}/G) = \bigcup_i \mathrm{Zero}(\mathrm{QD}(\mathrm{ASC}_i)/G).$$

后一个分解在某些应用,例如几何定理证明中更方便.详见[3].

上述定理实际上给出了代数簇的不可约分解与单纯维数分解,但其中有些分支是多余的.要想去掉这些多余分支,需要计算 QD(ASC) 与 PD(ASC) 的生成基.对于不可约升列 ASC,[4] 中给出了基于 Chow 坐标的计算方法.下面介绍一个相对简单的方法.

引理 4 对于任意升列 $\mathrm{ASC} = \{A_1, A_2, \cdots, A_k\}$,则有

$$\mathrm{QD}(\mathrm{ASC}) = \mathrm{Ideal}(A_1, A_2, \cdots, A_k, z_1A_1 - 1, z_2A_2 - 1, \cdots, z_kA_k - 1) \cap k[x_1, x_2, \cdots, x_k],$$

其中 z_i 是新变量[2].

根据以上结果,可以用 Groebner 基方法计算 QD(ASC) 的一组生成基(见文献[5]、[2]).最后有

定理 7(代数簇的唯一分解[1]) 对任意多项式的非空集合 PS 和多项式 G,可以求得不可约升列 ASC_i 与素理想 $PD(ASC_i)$ 的生成基 PS_i,使得

$$\mathrm{Zero}(\mathrm{PS}/G) = \bigcup_i \mathrm{Zero}(\mathrm{PD}(\mathrm{ASC}_i)/G) = \bigcup_i \mathrm{Zero}(\mathrm{PS}_i/G),$$

并且以上分解中任何一个分支都不能去掉.

1.2　复数域上的投影定理

大家知道,两个方程

$$f = x^n + a_1x^{n-1} + \cdots + a_n = 0,\quad g = x^m + b_1x^{m-1} + \cdots + b_m = 0$$

有公共解的条件是它们的下面结式为零,

$$R(f,g) = \begin{vmatrix} 1 & a_1 & \cdots & a_n & & \\ & \ddots & & & \ddots & \\ & & 1 & a_1 & \cdots & a_n \\ 1 & b_1 & \cdots & b_m & & \\ & \ddots & & & \ddots & \\ & & 1 & b_1 & \cdots & b_m \end{vmatrix}.$$

下面的投影定理可以看作这一概念的推广.设 PS 是 $K[U,x]$ 中的多项式集合,这里 U 代表变量 $\langle u_1,u_2,\cdots,u_m\rangle$,$x$ 代表 $\langle x_1,x_2,\cdots,x_n\rangle$.$D$ 是其中多项式,E 是 K 的代数闭包,定义拟代数集为

$$\mathrm{Zero}(\mathrm{PS}/D) = \{a \in E^{m+n} \mid \forall P \in \mathrm{PS}, P(a) = 0; D(a) \neq 0\}.$$

定义投影的概念如下:

$$\mathrm{Proj}_x\mathrm{Zero}(\mathrm{PS}/D) = \{e \in E^m \mid \exists a \in E^n, 使得(e,a) \in \mathrm{Zero}(\mathrm{PS}/D)\}.$$

如果 $m = 0$,视 Zero(PS/D)是否为空集,定义 $\mathrm{Proj}_x\mathrm{Zero}(\mathrm{PS}/D)$ 为 T 或 F.不难看出,投影实际上给出了一个方程组对于变量 x_i 有解的条件(见[6]).

定理 8　拟代数集的投影是若干拟代数集的并.进一步,存在一个算法在有限步内求得下面分解

$$\mathrm{Proj}_x\mathrm{Zero}(\mathrm{PS}/D) = \bigcup_{i=1}^{s}\mathrm{Zero}(\mathrm{AS}_i/H_iJ_i),$$

其中 AS_i 是 $K[U]$ 中的升列,H_i 是 $K[U]$ 中的多项式,J 是 AS_i 的初式之乘积.

为了给出上面定理的构造性证明,先引入下面引理,其证明见[6].

引理 5　设 $B = \sum_{i=0}^{t} B_ix^i \in K[U,x]$,其中 B_i 是 U 的多项式,则有

$$\mathrm{Proj}_x\mathrm{Zero}(\varnothing/B) = \bigcup_{i=0}^{t}\mathrm{Zero}(\varnothing/B_i).$$

引理 6　设 P、Q 是 $K[U,x]$ 中的多项式,$d = \deg_x P > 0$,$\deg_x Q > 0$,则有

$$\mathrm{Proj}_x\mathrm{Zero}(P/QI) = \mathrm{Proj}_x\mathrm{Zero}(\varnothing/RI).$$

其中 I 是 P 的初式,$R = \mathrm{prem}(Q^d,P)$.

算法 1

设 PS 是 $K[u_1,\cdots,u_m,x_1,\cdots,x_n]$ 中的多项式集合,D 是其中多项式,E 是 K 的代数闭包,计算 $\mathrm{Proj}_{x_1,\cdots,x_n}\mathrm{Zero}(\mathrm{PS}/D)$.

步 1　在变量顺序 $u_1 < \cdots, u_m < x_1 < \cdots < x_n$ 下应用吴-Ritt 分解定理,

$$\mathrm{Zero}(PS/D)=\bigcup_{i=1}^{s}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i).$$

对每个升列 ASC_i 进行下面操作.

步 2　设 ASC_i 形式如下：$\mathrm{ASC}_i=B_1,\cdots,B_r,A_1,\cdots,A_t$.其中 B_i 是变量 U 的多项式，A_i 中包含 x_i.不妨设 A_i 的主变元为 x_{n+i-t}.

步 3　首先计算 $Z=\mathrm{Proj}_{x_n}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i)$.如果 D 中不包含 x_n，则 $Z=\mathrm{Zero}(ASC'_i/DJ_i)$，其中 $\mathrm{ASC}'_i=B_1,\cdots,B_r,A_1,\cdots,A_{t-1}$.如果 D 包含 x_n，由引理 6，$Z=\mathrm{Proj}_{x_n}\mathrm{Zero}(\mathrm{ASC}'_i/RJ_i)$，其中 $R=\mathrm{prem}(D^d,A_t)$，$d=\deg_{x_n}A_t$.

步 4　重复步 3，可以得到

$$\mathrm{Proj}_{x_{n-t+1},\cdots,x_n}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i)=\bigcup_k\mathrm{Zero}(\{B_1,B_2,\cdots,B_r\}/G_k),$$

其中 G_k 是 $B_1,B_2,\cdots,B_r$ 的初式与 $K[U,x_1,x_2,\cdots,x_{n-t}]$ 中一个多项式 H_k 的乘积.

步 5　利用引理 6 将 H_k 中的变量 $x_1,x_2,\cdots,x_{n-t}$ 销去，最后得到所求的投影

$$\mathrm{Proj}_{x_1,\cdots,x_n}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i)=\bigcup_k\mathrm{Zero}(\{B_1,B_2,\cdots,B_r\}/F_k),$$

其中 F_k 是 B_k 的初式与一个 U 的多项式的乘积.

作为投影定理的一个应用，可以给出复数域上的一阶谓词逻辑公式的判定算法.复数域 $\mathbf{C}$ 上的**一阶谓词逻辑公式**(简称公式)可以严格定义如下：

(1) 复数域 $\mathbf{C}$ 上的多项式等式 $P(x_1,x_2,\cdots,x_n)=0$ 是一个公式.

(2) 如果 f、g 是两个公式，则 $f\wedge g$(与)，$f\vee g$(或)，$\neg f$(非) 也是公式；

(3) 如果 f 是一个公式，则 $\exists x_i\in\mathbf{C}(f)$，$\forall x_i\in\mathbf{C}(f)$ 也是公式.

下面是一些公式的例子，

$\forall c_1,\cdots,\forall c_n\exists x(x^n+c_1x^{n-1}+\cdots+c_n=0)$，即代数学基本定理.

$\forall x_1,\cdots,\forall x_n[(h_1=0\wedge\cdots h_s=0)\Rightarrow c=0]$，即定理证明.

设 f 是一个公式，通过一些简单的逻辑变换可以将 f 变为下面标准形式：

$$f=Q_1x_{s_1}\cdots Q_kx_{s_k}[\bigvee_{i=1}^{t}(P_{i1}=0\wedge\cdots\wedge P_{in_i}=0\wedge D_{i1}\neq0\wedge\cdots\wedge D_{im_i}\neq0)],$$

其中 Q 是谓词 $\exists$ 或者 $\forall$，P_{ij}、D_{ij} 是多项式.变量 $x_{s_1},\cdots,x_{s_k}$ 称为被限制的变量.未被限制的变量称为自由变量.所谓复数域 $\mathbf{C}$ 上的一阶谓词逻辑公式的判定是指下面两类问题：

(1) **判定问题**　如果 f 中无自由变量，判定公式 f 在复数域上是否正确.

(2) **谓词消去**　如果 f 中有自由变量，找到一个只含自由变量的公式 g，使得 f 与 g 在复数域上等价.

定理 9　复数域 $\mathbf{C}$ 上的一阶谓词逻辑公式的判定问题与谓词消去问题都可以由投影定理解决.

很明显，判定问题是谓词消去的特殊情形，因此只需要考虑谓词消去问题.不失一般性，设 $Q_s=\exists$，$s_k=n$.下面讨论怎样消去谓词($\exists x_n$).

设 $\mathrm{PS}=\{P_{i1},\cdots,P_{in_i}\}$，$D=\prod_{k=1}^{m_i}D_{ik}$，由算法 1，

$$\mathrm{Proj}_{x_n}\mathrm{Zero}(PS/D) = \bigcup_k \mathrm{Zero}(AS_k/F_k).$$

由投影的定义可以看出，逻辑公式

$$\exists x_n(P_{i1} = 0 \wedge \cdots \wedge P_{in_i} = 0 \wedge D_{i1} \neq 0 \wedge \cdots \wedge D_{im_i} \neq 0)$$

等价于

$$\bigvee_k (AS_k = 0 \wedge F_k \neq 0).$$

也说是说，已经从 f 中消除了量词($\exists x_n$)．再注意到：$\exists x(f \vee g) \Leftrightarrow \exists xf \vee \exists xg$，则有

$$f = Q_1 x_{t_1} \cdots Q_{k-1} x_{t_{k-1}} \exists x_n[\bigvee_{i=1}^{t}(f_i)] \Leftrightarrow Q_1 x_{t_1} \cdots Q_{k-1} x_{t_{k-1}}[\bigvee_{i=1}^{t}(\exists x_n(f_i))].$$

现在，已经可以从 f 中消除了量词($\exists x_n$)．如果 Q_k 不是 $\exists$，可以作下面的变换：

$$f = Q_1 x_{t_1} \cdots Q_{k-1} x_{t_{k-1}} \forall x_n[g] \Leftrightarrow Q_1 x_{t_1} \cdots Q_{k-1} x_{t_{k-1}} \neg \exists x_n[\neg g].$$

经过逻辑变换，不难将 $\neg g$ 变为标准形式．然后再用与上面相似的过程消去($\exists x_n$)．重复上面的过程即可完成谓词消去．

1.3　代数簇的各种表示及转换

代数簇有很多表示形式，这些表示形式都有其特定的意义与应用，本节将介绍代数簇的若干种表示形式及其互相之间的转换形式．

代数簇最直接的表示形式是生成元或隐式表示．由希尔伯特(D.Hilbert)有限基定理可知，对任一代数簇 V 一定存在有限多项式集合 $\mathrm{PS} \subset K[x_1, x_2, \cdots, x_n]$ 使得

$$V = \mathrm{Zero}(\mathrm{PS}). \tag{1-1}$$

代数簇的另一种常用的表示是参数表示．大家知道，

$$x = \frac{t^2-1}{t^2+1}, \quad y = \frac{2t}{t^2+1}$$

是单位圆的一组参数方程．一般地讲参数方程是指形式为

$$x_1 = \frac{P_1(u_1, \cdots, u_m)}{Q_1(u_1, \cdots, u_m)}, \cdots, x_n = \frac{P_n(u_1, \cdots, u_m)}{Q_n(u_1, \cdots, u_m)} \tag{1-2}$$

的一组有理方程．不难证明，只有不可约代数簇才能有参数表示．不可约代数簇还有其它一些表示，如 Chow 坐标、母元、预解式等．

设 E 为数域 K 的扩域．对 E^n 中两点 m、z，m 称为 z 的母元或 z 称为 m 的子元，如果对 $K[x]$ 中任一多项式 $F(x)$，由 $F(m) = 0$ 可以推出 $F(z) = 0$．点 m 的所有子元的集合记为 $\mathrm{Spec}(m)$．不难证明 $\mathrm{Spec}(m)$ 是一个不可约代数簇．点 m 称为代数簇 V 的母元，如果 $V = \mathrm{spec}(m)$．于是有

定理 10　不可约代数簇且只有不可约代数簇才存在母元．

不难证明，对于由参数方程(1-2)式定义的代数簇，($P_1/Q_1, P_2/Q_2, \cdots, P_n/Q_n$)就是其一个母元)．但是一般代数簇的母元则不容易显示给出．实际上，母元可以由不可约升列显示给出．

设 ASC: $f_1, f_2, \cdots, f_p$ 为一不可约升列, $c_i = \mathrm{cls}(f_i)$, 重新命名变量: $y_i = x_{c_i}$, 其它变量命名为 $u_1, u_2, \cdots, u_q$, $p + q = n$. 则 ASC 可以表示为

$$\mathrm{ASC}: f_1(u, y_1), f_2(u, y_1, y_2), \cdots, f_p(u, y_1, \cdots, y_p). \tag{1-3}$$

将变量 u 代换为一组自由变量后可以依次求得 $y_1, y_2, \cdots, y_p$, 从而得到 ASC 的一组解 a, 称为升列 ASC 的母元. 于是有

定理 11 设不可约升列的母元为 $m(\mathrm{ASC})$. 则

$$\mathrm{Zero}(\mathrm{PD}(\mathrm{ASC})) = \mathrm{Spec}(\mathrm{m}(\mathrm{ASC})).$$

也就是说, 任意不可约代数簇的母元均可表示为一个不可约升列的母元. 所以, 可以用不可约升列作为母元的构造型表示.

定理 12 设 ASC 是一个形如(1-3)的不可约升列. 可以求得整数 $c_1, c_2, \cdots, c_p$ 与新变元 $w = c_1y_1 + \cdots + c_py_p$, 使得在变元顺序 $u < w < y_1 < \cdots < y_p$ 下重新使用整序算法有

$$\mathrm{Zero}(\mathrm{ASC} \cup \{w - c_1y_1 - \cdots - c_py_p\}/J = \mathrm{Zero}(\mathrm{ASC}/J),$$

其中 J 是 AS 的初式的乘积. 则 ASC 具有下面形式

$$\begin{aligned}
&R(u_1, u_2, \cdots, u_q, w);\\
&I_1(u_1, u_2, \cdots, u_q, w)y_1 - U_1(u_1, u_2, \cdots, u_q, w),\\
&\cdots\cdots,\\
&I_1(u_1, u_2, \cdots, u_q, w)y_p - U_p(u_1, u_2, \cdots, u_q, w),
\end{aligned}$$

其中 R、I_i、U_i 均为变量 u 与 w 的多项式.

称 R 是 ASC 及素理想 PD(ASC) 的**预解式**. 由此得到不可约代数簇的预解式表示:

$$\begin{cases} R(u_1, u_2, \cdots, u_q, w); \\ y_1 = \dfrac{I_1(u_1, u_2, \cdots, u_q, w)}{U_1(u_1, u_2, \cdots, u_q, w)}, \cdots, y_p = \dfrac{I_p(u_1, u_2, \cdots, u_q, w)}{U_p(u_1, u_2, \cdots, u_q, w)}. \end{cases} \tag{1-4}$$

作为这一定理与吴-Ritt 零点分解定理的推论, 有

推论 2 设 $V = \mathrm{Zero}(\mathrm{PS})$ 为不可约代数簇, PS 为有限集合. 可以找到一个多项式 $P = 0$ 定义的超曲面 S, 使得 V 与 S 双有理等价, 且可以给出 S 与 V 之间的双有理变换.

用定理 12 中的符号, 由吴-Ritt 零点分解定理: $V = \mathrm{Zero}(\mathrm{PS}) = \mathrm{Zero}(\mathrm{PD}(\mathrm{ASC}))$, 设 R 为 V 的预解式. 则 V 到 S 与 S 到 V 的双有理变换如下.

$$(u_1, u_2, \cdots, u_q, y_1, y_2, \cdots, y_p) \Rightarrow (u_1, u_2, \cdots, u_q, c_1y_1 + \cdots + c_py_p),$$

$$(u_1, u_2, \cdots, u_q, w) \Rightarrow (u_1, u_2, \cdots, u_q, \frac{I_1}{U_1}, \cdots, \frac{I_p}{U_p}).$$

现在已给出了代数簇的四种表示方法. 下面介绍这四种表示之间的互相转换.

1. 由(1-1)式→(1-2)式

又称为隐式代数簇的参数化. 这一问题一般分为两步:

(1) 由推论 2, 可以找到一个多项式 $P = 0$ 定义的超曲面, 使其与给定的代数簇双有理等价, 并给出相应的双有理变换. 这样, 一般代数簇的参数化问题就变成

了超曲面的参数化问题.

(2) 寻找超曲面的参数方程.这一问题还远未彻底解决,只对曲线、曲面给出了存在算法[7、8、9].下面介绍[8]中给出的曲线的参数化方法.

现在要求平面曲线 $R(x,w)=0$ 的参数方程.最简单的方法是利用下面由 Luroth 定理推得的结果:若次数为 d 的平面曲线存在一组有理参数方程,则该曲线一定有一组次数为 d 的参数方程.设 $R(x,w)$ 的次数是 d,则令

$$x=P(u)/R(u),\ w=Q(u)/R(u).$$

其中 P、Q、R 是次数为 d 系数是未定元的多项式.将 P/R、Q/R 代入 $R(x,w)=0$,得到 P、Q、R 系数的一组多项式方程.可以利用吴方法来求这组方程的解,从而求得 P、Q、R.

2.由(1-2)式⟶(1-1)式

又称为参数方程的隐式化.对于一参数方程 C 的隐式化可以提出下面问题:

(1) 求参数方程定义的素理想的基.即以$(P_1/Q_1,P_2/Q_2,\cdots,P_n/Q_n)$为母点的素理想的基.

(2) 参数方程在 n 维空间形成的映象

$$\mathrm{IM}(C)=\{e\in E^n\mid e=(\frac{P_1(f_1,\cdots,f_m)}{Q_1(f_1,\cdots,f_m)},\cdots,\frac{P_n(f_1,\cdots,f_m)}{Q_n(f_1,\cdots,f_m)}),\quad f_i\in E\}$$

定义方程是怎样的?

(3) 参量 $u_1,u_2,\cdots,u_m$ 是否独立?如果不独立,是否能找到一组新的参数方程定义相同的代数簇且有独立的参量.

(4) 对于一组可能的 x 的值,相应的 u 的值是否唯一?如果不唯一,能否找到一组新参数方程与原方程等价且这样的值是唯一的.

(5) 设参数方程定义的素理想为 I.如果 $\mathrm{IM}(C)=\mathrm{Zero}(I)$,即参数方程的映像是一个代数簇,则称参数方程为正则.对于给定的参数方程,能判定其是否为正则?如果不是正则,需要找到一组正则参数方程与原参数方程定义相同的代数簇.

除第五个问题的最后一问,其它问题基于吴方法的解答见[8、10、11].下面给出前两个问题的解答.

定理 13 设 $F_i=P_i-y_iQ_i$, $G_i=z_iQ_i-1$, $i=1,2,\cdots,n$,其中 z_i 为新变元,则参数方程定义的素理想为 $\mathrm{Idea}(F_1,F_2,\cdots,F_n,G_1,G_2,\cdots,G_n)\cap K[y_1,y_2,\cdots,y_n]$.

基于上述定理,就可以用 Groebner 基([5])计算这一素理想的基.

定理 14 对于参数方程(C),可以找到其映象的下列唯一表示.

$$\mathrm{IM}(C)=\mathrm{Zero}(\mathrm{PD}(\mathrm{AS}))-\bigcup_{i=1}^{k}\mathrm{Zero}(\mathrm{AS}_i/H_i),$$

其中 AS、AS_i, $i=1,2,\cdots,k$,是不可约升列,H_i 是 AS_i 的初式与一个多项式的乘积,且 $\mathrm{Zero}(\mathrm{AS}_i/H_i)\subset\mathrm{Zero}(\mathrm{PD}(\mathrm{AS}))$.

(1-1)式→(1-3)式可以用吴零点分解定理来实现.

3.由(1-3)式→(1-4)式

设 ASC 是一个形如(1-3)的不可约升列.引进新变元 $w=c_1y_1+\cdots+c_py_p$,其中 c_i 是随机选取的整数.有([3])

定理 15　有算法可以求得整数 c_i，并在变元顺序 $u < w < y_1 < \cdots < y_p$ 下重新使用整序算法

$$\mathrm{Zero}(\mathrm{AS} \cup \{w - c_1 y_1 - \cdots - c_p y_p\}/I) = \mathrm{Zero}(\mathrm{AS}'/I),$$

其中 I 是 AS 的初式的乘积. 则 AS′ 具有下面形式.

$$R(u_1, u_2, \cdots, u_q, w),$$
$$R_1(u_1, u_2, \cdots, u_q, y_1) = I_1(u_1, u_2, \cdots, u_q)y_1 - U_1(u_1, u_2, \cdots, u_q),$$
$$\cdots\cdots$$
$$R_p(u_1, u_2, \cdots, u_q, y_p) = I_p(u_1, u_2, \cdots, u_q)y_p - U_p(u_1, u_2, \cdots, u_q),$$

其中 R 称为 AS 的预解式. 若随机选取 c_i，则 AS′ 具有上面形式的概率为 1.

4. 由 (1-4) 式→(1-3) 式

由定理 15 不难看出 $\{\mathrm{AS} \cup \{w - c_1 y_1 - \cdots - c_p y_p\}\}$ 与 AS′ 只不过是一个素理想在不同变量次序下的两个升列. 所以只需将 $R, R_1, \cdots, R_p$ 在变元顺序 $u < y_1 < \cdots < y_p < w$ 下重新使用整序算法即可得到 ASC.

1.4　奇异曲面的陈省身示性类

对一般代数簇，Grothendieck 引进了陈省身示性类（以下简称陈示性类）的概念，但须假定代数簇没有奇点. 如何在具有奇点的代数簇上定义陈示性类，自然成为数学界极为关注的难题. 利用不可约代数簇母元的概念，吴文俊对具有任意奇点的代数簇成功地建立了陈示性类的概念（[12、13]）. 当代数簇光滑时，吴给出的定义就是通常的陈示性类，而且他所定义的陈示性类是代数等价类，是具体可计算的.

对于具有奇点的代数簇 M，设 M^0 是 M 的一确定子簇，其包含 M 的全部奇点. 吴文俊建立了 M 上相对于 M^0（模掉 M^0）的不可约代数等价系的概念. 由 M 上 c 维不可约子簇所定的代数等价系在自然加法之下构成加法群，记为 $\mathrm{ALG}_c(M/M^0)$. 而通常意义下 c 维代数等价系所构成的加法群记为 $\mathrm{ALG}_c(M)$. 由于一个不可约代数等价系中的元素都属于同一代数等价系，因之有一自然同态

$$J_c : \mathrm{ALG}_c(M/M^0) \to \mathrm{ALG}_c(M). \tag{1-5}$$

现设 M 是复投影空间 CP_m 中 d 维不可约代数簇，M^0 是包含 M 的全部奇点的 M 的子簇. 再设 N 是另一复投影空间 CP_n 中的 d 维不可约代数簇，N^0 为包含 N 全部奇点的 N 的子簇. 假设 T 是从 N 到 M 的双有理变换，满足下述条件：

1° T 在 N 上处处有定义；

2° $T(x) \in M^0$，当且仅当 $x \in N^0$；

3° T 在 N/N^0 上为一双射.

则可证明，双有理变换 T 诱导出同态

$$T_c : \mathrm{ALG}_c(N/N^0) \to \mathrm{ALG}_c(M/M^0). \tag{1-6}$$

如果 V_n 为复投影空间 CP_n 中的 n 维没有奇点的不可约子簇，在通常意义下可在 V_n 上定义交截. 设 N 是 V_n 上 b 维不可约子簇，N^0 是 N 在 V_n 上确定的子簇，其包含 N

的全部奇点. 对任意代数等价系 $S \in \mathrm{ALG}_a(V_n)$, 则 S 中任意与 N 和 N^0 都简单相交的元素 U_a 相交后所得代数簇属于另一代数等价系, 由此可诱导出同态

$$I_a: \mathrm{ALG}_a(V_n) \to \mathrm{ALG}_{a+b-n}(N/N^0). \tag{1-7}$$

利用公式(1-5)、(1-6)、(1-7)给出的三个同态映射可定义复投影空间 CP_n 中 d 维不可约代数簇 V_d 的 Ehresmann 类. 设 V_d^0 是 V_d 的包含其全部奇点的子簇. 对给定的维数 s, 有同态

$$J_s: \mathrm{ALG}_s(V_d/V_d^0) \to \mathrm{ALG}_s(V_d). \tag{1-8}$$

设 z 是 V_d 的母点, L_z 是 V_d 在点 z 的切空间. 则 (z, L_z) 作为一母点, 可在复合 Grassmann 簇 $\mathrm{GR}(n;0,d)$ 中确定一 d 维 Schubert 簇 W_d, 由于 $\mathrm{GR}(n;0,d)$ 可视为充分高维数 N 的复投影空间 CP_n 中的子簇, W_d 中任一点 (x, L_x) 都是 (z, L_z) 的特定化, 于是 x 是 V_d 中确有定义的点. 若命映射 $T:(x, L_x) \to x$, 则有

定理 16　映射 T 是从 W_d 到 V_d 的双有理变换, 且 T 满足前面述及的性质 1°, 2°, 3°.

依此定理可有同态

$$T: \mathrm{ALG}_a(W_d/W_d^0) \to \mathrm{ALG}_a(V_d/V_d^0). \tag{1-9}$$

如上所述, $\mathrm{GR}(n;0,d)$ 是复投影空间的子簇, 所以它的任一子簇都代数等价于与 W_d 和 W_d^0 都简单相交的子簇, 故有

$$I_s: \mathrm{ALG}_s(\mathrm{GR}(n;0,d)) \to \mathrm{ALG}_t(W_d/W_d^0), \tag{1-10}$$

其中 $t = s + d - \dim\mathrm{GR}(n;0,d)$. 对于 $\mathrm{GR}(n;0,d)$ 又有对偶映射

$$\mathrm{Dual}: \mathrm{ALG}_s(GR(n;0,d)) \to \mathrm{ALG}_{s'}(\mathrm{GR}(n;0,d)),$$

其中 $s' = \dim GR(n;0,d) - s$. 现取 Ehresmann 符号

$$\mathrm{EH} = [a_0 \mid b_0, b_1, \cdots, b_d], s = \sum{}'(b_k - k) + a_0,$$

其中 $\sum{}'$ 号表示对 k 从 0 到 d 求和, 但要去掉 $b_k = a_0$ 的项, 则有如下定义

定义 5　设 $r = d - s$. 代数等价类 J_r、T、$I_{s'}$、Dual、$EH \in \mathrm{ALG}_r(V_d)$, 称为 V_d 的对应于符号 EH 的 Ehresmann 类, 记为 $\mathrm{EH}(V_d) = [a_0 \mid b_0, b_1, \cdots, b_d](V_d)$.

在这些 Ehresmann 类中有一些特别重要.

定义 6　代数簇 V_d 的一类特定的 Ehresmann 类称为 Gamkrelidze 类, 记为 $\mathrm{GM}_{st} = [s - t \mid (0, \cdots, d - t), (d - t + 2), \cdots, d + 1)](V_d) \in \mathrm{ALG}_r(V_d)$, 其中 $r = d - s$.

据此, 吴文俊引进了具有奇点代数簇的陈示性类的定义.

定义 7　具有奇点的代数簇 V_d 的陈示性类定义为若干 Gamkrelidge 类的组合

$$\mathrm{CH}_s(V_d) = \sum_{t=s}^{d}(-1)^t \binom{d - t + 1}{d - s + 1} G_{st}(V_d) \in \mathrm{ALG}_r(V_d),$$

其中 $r = d - s$. 当 $s = d$, 则陈示性类 $\mathrm{CH}_d(V_d) \in \mathrm{ALG}_0(V_d)$ 为一整数, 因其是 Severi 意义下的投影特征数, 故称为代数簇 V_d 的陈特征数(参阅文献[14]). 这一定义的合理性由下面的定理所阐明.

定理 17　如上所定义的代数簇的陈示性类, 当代数簇光滑时就是通常意义下

的陈示性类.

下面介绍具有奇点超曲面的陈特征数的性质.特别是给出了有关陈特征数的下列丘成桐-Miyaoka不等式的推广

$$3c_2 - c_1^2 \leqslant 0. \tag{1-11}$$

对于具有负纯曲率的 n 维紧 Kaehler-Einstein 流形 M,丘成桐还证明了其陈特征数满足

$$(-1)^n[2(n+1)c_2(M)c_1^{n-2}(M) - nc_1^n(M)] \geqslant 0 \tag{1-12}$$

并且证明这一不等式对具有丰富典范丛的 n 维代数簇依然成立.他问道:关于 M 是否还存在这种类型的陈特征数不等式?

设 H_n 为 $n+1$ 维复投影空间中具有任意奇点的超曲面.文[15、16]证明了

定理 18 对任意给定的正数 $k, 2 \leqslant k \leqslant n$,具有任意奇点的超曲面 H_n 有

$$(-1)^k \sum_{i=1}^{k} l_i^{(k)} \mathrm{CH}_{hi(k)}(H_n) Q^{n-k} H_n \geqslant 0. \tag{1-13}$$

特别当 $k = n$ 时有

$$(-1)^n \sum_{i=1}^{n} l_i^{(n)} \mathrm{CH}_{hi(n)}(H_n) \geqslant 0. \tag{1-14}$$

其中 l_i 由下式给定:

$$l_1^{(k)} = (-1)^{k-1} \frac{n}{k} \binom{n+1-i}{n-k+1}, \quad l_i^{(k)} = (-1)^{k-i}(n+1)^{i-1} \binom{n+1-i}{n-k+1},$$
$$i = 2, 3, \cdots, k. \tag{1-15}$$

由于 k 和 n 可任意给定,这一定理给出了本节开始所提出的前两个问题的系统解答.依据定义7,文[14,15]证明:具有任意奇点的超曲面的陈示性类满足等式关系.当 $k \geqslant 4$,在 k 的分拆中存在非勾形分拆.现考虑 $2 \leqslant k \leqslant n$,由具有奇点的超曲面 H_n 的对应于 k 的分拆 $p(k)$ 的全体陈示性类所张成的线性空间为

$$\mathrm{CH}^{(k)}(H_n) = \{\mathrm{CH}_{p(k)}(H_n) \mid p(k) \text{ 是 } k \text{ 的所有分拆.}\}$$

定理 19 对具有奇点的 n 维超曲面 H_n 及任意给定的 $k, 2 \leqslant k \leqslant n$,线性空间 $\mathrm{CH}^{(k)}$ 的维数等于 k.

由于当 $k \geqslant 4$ 时 k 的分拆个数多于 k,因此定理19的直接推论是

推论 3 对具有任意奇点的 n 维超曲面 H_n,当 $n \geqslant 4$ 时,H_n 的陈示性类满足一系列等式关系.

例:对 $4 = k \leqslant n$,H_n 的陈示性类满足下面的等式

$$a_0 \mathrm{CH}_{22}(H_n) + \sum_{i=1}^{4} a_i \mathrm{CH}_{hi(4)}(H_n) = 0, \tag{1-16}$$

其中 $a_0 = 3n^2 + 13n + 16$, $a_1 = -n(n+1)$, $a_2 = 4(n+2)^2$, $a_3 = -2(n^2-4)$, $a_4 = 2(n+1)(n+2)$.

利用推论3中得到的陈示性类的等式和公式(1-13)给出的陈特征数不等式进行组合,又可得到一系列新的陈特征数不等式.伴随着数 k 的增大,陈示性类所满足的等式个数将急剧增加.例如当 $k = 6$ 时,理论上 H_n, $n \geqslant 6$ 的陈示性类所满足的

等式数目将增加到330.

2　构造性微分代数几何

上面已经介绍了吴方法的代数形式.事实上这一套理论可以推广至微分情形.本节将简单地介绍微分域上的吴方法及其在相关问题中的应用.详细论述请见[17、18].

2.1　微分域上的吴-Ritt零点分解定理

一个域 K 称作微分域,如果域 K 中除了具有通常域的运算与性质外,还有一个微分算子D满足:

$$\mathrm{D}(a+b)=\mathrm{D}a+\mathrm{D}b,$$
$$\mathrm{D}(ab)=\mathrm{D}ab+a\mathrm{D}b.$$

例如有理函数域 $\mathbf{Q}(t)$ 在微分算子 $\mathrm{D}=\mathrm{d}/\mathrm{d}t$ 下就构成一微分域.一个微分域 E 称作另一微分域 K 的一个**微分扩域**,如果域 E 是 K 的扩域且 E 的微分算子局限在 K 上也正好是 K 的微分算子.

设 $x_1,x_2,\cdots,x_n$ 是 n 个固定的变元,用 $x_{i,j}$ 表示变元 x_i 的 j 阶微分,K 上任何一个以 $x_{i,j}$ 为变元的多项式都称为 $x_1,x_2,\cdots,x_n$ 的微分多项式.所有 K 上微分多项式的全体构成的集合用 $K\{x_1,x_2,\cdots,x_n\}$ 表示.像通常吴方法中对多项式的讨论一样,对微分多项式也可以引入诸如类、秩、升列,伪除等概念,并进行类似的讨论.

设 P 为一微分多项式,P 的类 $\mathrm{cls}(P)$ 是指 P 中实际出现的变元的最大的下标.如果 P 属于 K,则定义 $\mathrm{cls}(P)=0$.对任一变元 x_i,P 中实际出现的变元 x_i 的微分的最高阶称为 P 对于 x_i 的阶,记作 $ord(P,x_i)$.当 x_i 及其微分不在 P 中出现时,则约定 $\mathrm{ord}(P,x_i)=-1$.

对两个微分多项式 P_1 和 P_2,称 P_2 比 P_1 有较高的秩,记作 $P_2>P_1$ 或 $P_1<P_2$ 如果:

1° $\mathrm{cls}(P_2)>\mathrm{cls}(P_1)$,

或

2° $\mathrm{cls}(P_2)=\mathrm{cls}(P_1)=p$ 且 $\mathrm{ord}(P_2,x_p)>\mathrm{ord}(P_1,x_p)$,

或

3° $\mathrm{cls}(P_2)=\mathrm{cls}(P_1)=p,\mathrm{ord}(P_2,x_p)=\mathrm{ord}(P_1,x_p)=r$ 且 $\deg_{x_{p,r}}P_2>\deg_{x_{p,r}}P_1$.

当两微分多项式 P_1 和 P_2 不能比较秩的高低时,称 P_1 和 P_2 有相同的秩,记作 $P_1\sim P_2$.假设微分多项式 P 的类 $\mathrm{cls}(P)=p>0,\mathrm{ord}(P,x_p)=m$.则 P 和 DP 可以分别写成

$$P=a_d x_{p,m}^d+a_{d-1}x_{p,m}^{d-1}+\cdots+a_0,$$

$$\mathrm{D}P = Sx_{p,m+1} + \mathrm{D}a_d x_{p,m}^{d-1} + \cdots + \mathrm{D}a_0,$$

其中 $d = \deg_{x_{p,m}} P$，$a_d \neq 0$，a_i 是比 $x_{p,m}$ 有较低秩的微分多项式．a_d 称为 P 的初式，记作 $\mathrm{Init}(P)$，$S = da_d x_{x_{p,m}}^{d-1} + \cdots + a_1$ 叫做 P 的**隔离子**，记作 $\mathrm{Sep}(P)$．

设 P 是一微分多项式，$\mathrm{cls}(P) = p > 0$．一微分多项式 Q 称为对 P 已约化，如果 $\mathrm{ord}(Q, x_p) < \mathrm{ord}(P, x_p)$ 或 $r = \mathrm{ord}(P, x_p) = \mathrm{ord}(Q, x_p)$ 且 $\deg_{x_{p,r}} Q < \deg_{x_{p,r}} P$．如果一微分多项式 G 对 P 不是约化的，则可以对 G 可进行如下约化．

伪除算法：

首先令 $m = \mathrm{ord}(P, x_p)$，$S = \mathrm{Sep}(P)$，$I = \mathrm{Init}(P)$．

步 1　如果 $\mathrm{ord}(G, x_p) = m$，则将 G 和 P 都看作一般意义下 $x_{p,m}$ 的多项式，并按照 1.1 节的方法对 G 用 P 进行约化．

步 2　如果 $\mathrm{ord}(G, x_p) = n > m$，令 $k_1 = n - m$，则 P 的 k_1 阶微分 $P^{(k_1)}$ 是 $x_{p,n}$ 的线性多项式，并以 S 为其初式．将 G 和 $P^{(k_1)}$ 都看成一般意义下 $x_{p,n}$ 的多项式，并利用 1.1 节的方法对 G 用 $P^{(k_1)}$ 进行约化，则可以找到最小的非负整数 v_1 与微分多项式 C_1 和 G_1，满足

$$S^{v_1} G = C_1 P^{(k_1)} + G_1, \quad \mathrm{ord}(G_1, x_p) < n.$$

然后对 G_1 重复步 1 或步 2 的过程．

一般来说，有：

定理 1　对任意两个微分多项式 P 和 G，$\mathrm{cls}(P) \neq 0$，则总可找到一组非负整数 v、u 和 $k_1, k_2, \cdots, k_s$ 以及微分多项式 $Q_1, Q_2, \cdots, Q_s, Q, R$，使得

$$S^v I^u G = Q_1 P^{(k_1)} + Q_2 P^{(k_2)} + \cdots + Q_s P^{(k_s)} + QP + R,$$

其中 $S = \mathrm{Sep}(P)$，$I = \mathrm{Init}(P)$，且 R 比 P 有较低的秩．

上述定理中的 R 称为 G 对 P 的余式，用 $\mathrm{prem}(G, P)$ 表示．

定义 1　非零微分多项式的有限序列 $P_1, P_2, \cdots, P_r$ 称为一个升列，如果

1° $r = 1$ 且 $P_1 \neq 0$，

或

2° $0 < \mathrm{cls}(P_1) < \cdots < \mathrm{cls}(P_r)$，且对任意的 $i > j$，$\mathrm{Init}(P_i)$ 都对 P_j 已约化．

一微分多项式 G 对一升列 $P_1, P_2, \cdots, P_r$ 的余式，归纳地定义为

$$\mathrm{prem}(G, P_1, \cdots, P_r) = \mathrm{prem}(\mathrm{prem}(G, P_r), P_1, P_2, \cdots, P_{r-1}).$$

利用 1.1 节相同的方法，同样可以比较两微分多项式升列秩的高低，定义微分多项式组的基列等概念，这里不再重复．

设 E 是 K 的一个微分扩域，HS 是微分多项式组，D 是微分多项式，定义：

$$d\text{-Zero}(\mathrm{HS}) = \{z \in E^n \mid 对所有的\ P \in \mathrm{HS}, P(z) = 0\},$$

$$d\text{-Zero}(\mathrm{HS}/D) = d\text{-Zero}(\mathrm{HS}) - d\text{-Zero}(D).$$

d-Zero(HS) 中的元素称为 HS 的零点．

平行于 1.1 节中的定理 2—定理 4 有

定理 2　对任意一微分多项式的非空集合 HS，有一机械化方法在有限步内或判定出 d-Zero(HS) 为一空集或找出一升列 CS 满足

$$d\text{-Zero}(CS/J) \subset d\text{-Zero}(HS) \subset d\text{-Zero}(CS),$$

且对任意 $P \in HS$, $\mathrm{Prem}(P, CS) = 0$,其中 J 是由 CS 中微分多项式的初式与隔离子的乘积.

上述定理中的 CS 称为 HS 的特征列,求得 CS 的过程叫做微分多项组的整序.

定理3(零点分解定理——弱形式)　设 HS 是微分多项式的有限集,D 是一个微分多项式.则有一机械化方法可在有限步内或判定出 $d\text{-Zero}(HS/D)$ 为一空集或找到一组升列 ASC_i,使得

$$d\text{-Zero}(HS/D) = \bigcup_i d\text{-Zero}(ASC_i/DJ_i),$$

且对任意 $P \in DS$, $\mathrm{prem}(P, ASC_i) \neq 0$,其中 J_i 是 ASC_i 中微分多项式的初式与隔离子组成的集合.

升列 ASC 称为不可约升列,如果 ASC 作为一般意义下多项式的升列是不可约的.此时将所有的变元及其微分都看成通常意义下的变元,则有

定理4　若 ASC 是不可约升列,则有

1° PD(ASC) 是一个素理想,$d\text{-Zero}(PD(ASC))$ 是一个不可约微分代数簇,

2° 对任意微分多项式 P, $P = 0(\mathrm{mod})\mathrm{Spec}(G)$,当且仅当 P 对于 d-ASC 的余式为零.

定理5(零点分解定理——强形式)　与定理3的叙述相同,只是其中的 ASC_i 是不可约升列.

定理6(微分代数簇的分解定理)　设 HS 是微分多项式的有限集,则有一机械化方法可在有限步内将 $d\text{-Zero}(HS)$ 分解为不可约代数簇之并

$$d\text{-Zero}(HS) = \bigcup_i d\text{-Zero}(PD(ASC_i)),$$

其中 ASC_i 是不可约微分代数簇.

2.2　微分域上投影定理

设 PS 是 $K\{U, x_2\}$ 中的多项式集合,这里 $U = \{u_1, u_2, \cdots, u_m\}$, $x = \{x_1, x_2, \cdots, x_n\}$, D 是其中多项式,E 是 K 的微分闭包.定义投影的概念如下

$$\mathrm{Proj}_x\mathrm{Zero}(PS/D) = \{e \in E^m \mid \exists a \in E^n, 使得(e, a) \in \mathrm{Zero}(PS/D)\}.$$

如果 $m = 0$,视 Zero(PS/D) 是否为空集,定义 $\mathrm{Proj}_x\mathrm{Zero}(PS/D)$ 为 T 或 F.不难看出,投影实际上给出了一个方程组对于变量 x_i 有解的条件.我们有

定理7　拟代数集的投影是若干拟代数集的并.进一步,存在一个算法在有限步内求得下面分解

$$\mathrm{Proj}_x\mathrm{Zero}(PS/D) = \bigcup_{i=1}^{t} \mathrm{Zero}(AS_i/HI_i),$$

其中 AS_i 是 $K[U]$ 中的升列,H 是 $K[U]$ 中的多项式,I_i 是 AS_i 的初式之乘积.

为了给出上面定理的构造性证明,先引入下面引理.

引理1　设 $B = \sum_{i=0}^{t} B_i x^i \in K[U, x]$,其中 B_i 是 U 的多项式,则有

$$\mathrm{Proj}_x\mathrm{Zero}(\varnothing/B) = \bigcup_{i=0}^{t}\mathrm{Zero}(\varnothing/B_i).$$

引理 2 设 P、Q 是 $K[U,x]$ 中的多项式，$d = \deg_x P > 0, \deg_x Q > 0$，则有

$$\mathrm{Proj}_x\mathrm{Zero}(P/QI) = \mathrm{Proj}_x\mathrm{Zero}(\varnothing/RI).$$

其中 I 是 P 的初式，$R = \mathrm{prem}(Q^d, P)$.

算法 1 设PS是 $K[U,x]$ 中的多项式集合，D 是其中多项式，E 是 K 的代数闭包，将计算 $\mathrm{Proj}_x\mathrm{Zero}(\mathrm{PS}/D)$.

步 1 在变量顺序 $u_1 < \cdots < u_m < x_1 < \cdots < x_n$ 下应用吴-Ritt 分解定理.

$$\mathrm{Zero}(\mathrm{PS}/D) = \bigcup_{i=1}^{s}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i).$$

对每个升列 ASC_i 进行下面操作.

步 2 设 ASC_i 形式如下：$\mathrm{ASC}_i = B_1\cdots, B_r, A_1, \cdots, A_t$，其中 B_i 是变量 U 的多项式，A_i 中包含 x_i. 不妨设 A_i 的主变元为 x_{n+i-t}.

步 3 首先计算 $Z = \mathrm{Proj}_{x_n}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i)$. 如果 D 中不包含 x_n，则 $Z = \mathrm{Zero}(\mathrm{ASC}'_i/DJ_i)$，其中 $\mathrm{ASC}'_i = B_1, \cdots, B_r, A_1, \cdots, A_{t-1}$. 如果 D 包含 x_n，由引理 2，$Z = \mathrm{Proj}_{x_n}\mathrm{Zero}(\mathrm{ASC}'_i/RJ_i)$，其中 $R = \mathrm{prem}(D^d, A_t)$，$d = \deg(A_t, x_n)$.

步 4 重复步 3，可以得到

$$\mathrm{Proj}_{x_{n-t+1},\cdots,x_n}\mathrm{Zero}(\mathrm{ASC}_i/DJ_i) = \bigcup_k \mathrm{Zero}(\{B_1, B_2, \cdots, B_r\}/G_k),$$

其中 G_k 是 $B_1, B_2, \cdots, B_r$ 的初式与 $K[U, x_1, x_2, \cdots, x_{n-1}]$ 中一个多项式 H_k 的乘积.

步 5 利用引理 2 将 H_k 中的变量 $x_1, x_2, \cdots, x_{n-t}$ 销去，最后得到所求的投影

$$\mathrm{Proj}_x\mathrm{Zero}(\mathrm{ASC}_i/DJ_i) = \bigcup_k \mathrm{Zero}(\{B_1, B_2, \cdots, B_r\}/F_k),$$

其中 F_k 是 B_i 的初式与一个 U 的多项式的乘积.

与 1.2 节相似，利用投影定理可以给出复数域上的一阶谓词逻辑公式的判定算法.

2.3 偏微系统的完全可积理论

1.1 节的概念与结果均可推广到偏微情形. K 称为独立变量 $x_1, x_2, \cdots, x_m$ 的微分域，如果域 K 中除了具有通常域的运算与性质外，还有 m 个微分算子 D_i, $i = 1, 2, \cdots, m$ 满足：

1° $\mathrm{D}_i(x_i) = 1, \mathrm{D}_i(x_j) = 0, i \neq j$,

2° $\mathrm{D}_i(a + b) = \mathrm{D}_i(a) + \mathrm{D}_i(b)$,

3° $\mathrm{D}_i(ab) = \mathrm{D}_i(a)b + a\mathrm{D}_i(b)$,

4° $\mathrm{D}_i(\mathrm{D}_j(a)) = \mathrm{D}_j(\mathrm{D}_i(a))$.

为了方便，引入数组的概念. 对于一个 m 元有序数组 $t = (I_1, I_2, \cdots, I_m)$，定义 $C_i(t) = I_i$, $\mathrm{ord}(t) = I_1 + \cdots + I_m$. 数组 u 是 v 的整倍数记为 $u \gg v$. 所有 m 元有序数组记为 Tot. 对于 Tot 的任意子集 T, $\mathrm{Tot}(T)$ 是 T 中所有数组的整倍数的数组的集

合.两个数组的乘积定义为相应参数的乘积.可以在所有数组之间引入一个顺序：$u > v$，如果

1° $\mathrm{ord}(u) > \mathrm{ord}(v)$

或者

2° $\mathrm{ord}(u) = \mathrm{ord}(v)$ 且存在 k，使得 $C_k(u) > C_k(v), C_i(u) = C_i(v), i < k$.

一个数组称为**素组**，如果其中任意元素不是另一元素的倍数.对一个数组 T，

$$\mathrm{Max}(T) = (n\text{-}\mathrm{Max}(C_i(t)), \cdots n\text{-}\mathrm{Max}(C_m(t))), t \in T.$$

数组 T 的**完整化**定义为

$$\mathrm{Comp}(T) = \{u \mid u \ll \mathrm{Max}(T), u \gg t \in T\}.$$

对一个数组 T 与数组 $t \ll \mathrm{Max}(T)$，整数 i 称为 t 的乘子，如果 $C_i(t) = C_i(\mathrm{Max}(T))$.记 $\mathrm{Mult}(t/T)$ 为所有 t 对于 T 的乘子的集合，$\mathrm{Nult}(t/T)$ 为所有 t 对于 T 的非乘子的集合.

$$\mathrm{TMU}(t/T) = \{t \cdot u \mid C_i(u) = 0 \text{ 对 } \mathrm{Nult}(t/T) \text{ 中所有 } I\}.$$

有了以上记号，1.1 节的概念可以相应推广到偏微情形.设 t 为数组，用 $D_t(P)$ 记多项式 P 对于 x_i 的 $C_j(t)$ 阶微分.对于微分多项式 P，用 $\mathrm{Ld}(P)$ 记 P 中次序的最高微分，则 $\mathrm{ord}(P) = \mathrm{ord}(\mathrm{Ld}(P))$，$\mathrm{cls}(P)$ 等于 $\mathrm{Ld}(P)$ 的下标，$\deg(P)$ 等于 $\mathrm{Ld}(P)$ 在 P 中的次数，$\mathrm{Init}(P)$ 等于 $\mathrm{Ld}(P)$ 最高次幂中的系数，$\mathrm{Sep}(P)$ 等于 P 对于 $\mathrm{Ld}(P)$ 的形式微分.

第一章中的主要结果在微分情形都成立.例如有

定理 8　对任意微分多项式集合 PS，存在有限微分多项式集合 FPS，使得 $d\text{-Zero(PS)} = d\text{-Zero(FPS)}$.

定理 9　对任意不可约微分代数簇 $V = d\text{-Zero(PS)}$，其中 PS 为一素理想.PS 的特征基 ASC 的母元 z 就是 V 的母元，即 $V = \mathrm{Spec}(z)$.

下面主要介绍偏微方程组的可积理论.对于一个非平凡升列

$$d\text{-ASC}: \quad D_1, D_2, \cdots, D_r,$$

$\mathrm{Ld}(D_i)$ 称为**主微分**，$\mathrm{Ld}(D_i)$ 的微分称为**次主微分**.其它微分称为**参量微分**.引入下面符号.对于 $1 \leqslant p \leqslant n$

$$\mathrm{LTUP}_p(d\text{-ASC}) = \{t \mid D_t(Y_p) \text{ 是某些 } \mathrm{Ld}(D_i)\}.$$

$\mathrm{Max}_p(d\text{-ASC}) = (\mathrm{Max}\{c_1(t)\}, \cdots, \mathrm{Max}\{\mathrm{Cm}(t)\}), t \in \mathrm{CTUP}_p(d\text{-ASC})$.

$\mathrm{CTUP}_p(d\text{-ASC}) = \mathrm{Comp}(\mathrm{LTUP}_p(d\text{-ASC}))$.

形如 $D_t(Y_p)(t \in \mathrm{LTUP}_p(d\text{-ASC}))$ 的主微分称为 d-ASC 的 C 微分.对于 $\mathrm{CTUP}_p(d\text{-}ASC) \setminus \mathrm{LTUP}_p(d\text{-ASC})$ 中的数组 v，作余式：

$$J^* D_v(Y_p) = R_{vp} d\text{-mod}(d\text{-ASC}).$$

称 $J^* D_v(Y_p) = R_{vpd} - \mathrm{mod}(d - \mathrm{ASC})$ 是关于 v 与 p 的导出微分多项式.d-ASC 的所有导出多项式与 d-ASC 中多项式重新按升次排列称为 d-ASC 的**完备化**.

$$d\text{-ASC}+: H_1, H_2, \cdots, H_t,$$

d-ASC 的一个 M 微分是下列形式的一个微分多项式：$\mathrm{DM} = D_u(H_i)$，其中 $\mathrm{Ld}(H_i) = D_t(Y_p), t \in \mathrm{CTUP}_p; C_i(u) = 0, i \in \mathrm{Nult}(t/\mathrm{LTUP}_p)$ 或者 $u \in$

$TMU(t/LTUP_p)$. 若干 M 微分的乘积称为一个 M 乘积. 一个 M 多项式是 M 乘积的线性组合,其中的系数只包含主微分与参量微分.

假定一个微分多项式 P 中包含有 M 微分 M_h 与其它微分,设 $D_v(Y_p)$ 是次主微分中次序最高者. 可以将 v 唯一分解为 ut,其中 $t \in CTUP_p \setminus LTUP_p$,且对 $i \in Nult(t/LTUP_p)$,$C_i(u)=0$. 设 $D_t(Y_p)=L_d(H_k)$,则可以通过微分 H_k 将 $D_v(Y_p)$ 消去. 这一消去过程称为 ***M* 约化**.

假定前述微分多项式 P 中除 M 微分与参量微分外还有主微分. 可以通过对 d-ASC 的伪除法将之消去. 这一过程称为 ***I* 约化**.

通过 M 约化与 I 约化后,有

$$J^* P = M(P) + N(P)$$

满足:①J 是 d-ASC 的初式与分离式的乘积,②$M(P)$ 是一个 M 多项式,③$N(P)$ 是一个只含参量微分与主微分的多项式,④$M(P)$、$N(P)$ 中不是 M 微分的主微分的次数小于 d-ASC 中相应微分的次数. $M(P)$ 与 $N(P)$ 称为 P 的 M 与 N 部分.

假定 d-ASC + 中存在 H_h,其 $L_d(H_h)=D_t(Y_p)$,且 t 有一个非乘子 $i \in Nult(t/LTUP_p)$. 则有 $D_i(H_h)=S^* D_u(Y_p)+R$,其中 $D_u(Y_p)$ 是 d-ASC + 中另一个多项式 H_k 的主微分. 从 $D_i(H_h)$ 与 H_k 中消去 $D_u(Y_p)$ 得到一个微分多项式 W 其主微分低于 $D_u(Y_p)$. W 的 N 部分称为 H_h 与非乘子 i 的积分多项式. 一个非平凡的微分升列称为被动的,如果其所有积分多项式均为零.

通过引入泰勒形式级数,也可以定义不可约升列的母元,见文献[17]. 下面结果是至关重要的.

定理 10 若 d-ASC 是一个被动的不可约升列,G 是其母元,则有

1° $Spec(G)$ 是一个不可约微分代数簇,

2° 对任意微分多项式 P,$P(G)=0$ 当且仅当 P 对于 d-ASC 的余式为零,

3° 对任意微分多项式 P,$P=0 \pmod{Spec(G)}$,当且仅当 P 对于 d-ASC 的余式为零.

因被动升列与不可约升列的定义是构造性的,不难给出偏微情形下微分代数簇的零点分解定理与微分代数簇的分解定理,其形式与上节定理 3、定理 5、定理 6 相同.

2.4 微分几何与力学中的定理证明与发现

与初等几何的情形相似,微分多项式组的整序算法同样可以应用于微分几何中定理的机器证明与一些几何关系或公式的自动发现与推导. 详细论述请见文献[17、19、20、21、22]. 下面举例说明.

例 1 两平面曲线 C_1 和 C_2 称为平行的,如果 C_1 到 C_2 有一个 1-1 对应,使得对应点的连线恰好是 C_1、C_2 的公共法线. 证明对任意两条这样的曲线,对应点之间的距离是一常数.

证明 假定 P_1 和 P_2 是曲线 C_1 和 C_2 上相对应的点,并设 $P_1=(x_1(t),x_2(t))$,$P_2=(x_3(t),x_4(t))$ 是其坐标的参数表示,记 P_1 和 P_2 的距离 P_1P_2 为

$x_5(t)$,则定理的假设可表示为

$$h_1 = (x_3 - x_1)x'_1 + (x_4 - x_2)x'_2 = 0,$$
$$h_2 = (x_3 - x_1)x'_3 + (x_4 - x_2)x'_4 = 0,$$
$$h_3 = x_5^2 - (x_3 - x_1)^2 - (x_4 - x_2)^2 = 0.$$

结论可写为:$g = x'_5 = 0$.

通过对微分多项式集$\{h_1, h_2, h_3\}$进行整序,可得其升列 CS 为

$$(x'^2_1 + x'^2_2)x'_2x'_3 + (x_3 - x_1)x'_1(x''_1x'_2 - x'_1x''_2) - x'_1x_2(x'^2_1 + x'^2_2),$$
$$x'_2x_4 + (x_3 - x_1)x'_1 - x_2x'_2,$$
$$x_5^2 - (x_3 - x_1)^2 - (x_4 - x_2)^2.$$

可以看出其初式与隔离子只有因子 $d_1 = x'_2, d_2 = x_3 - x_1, d_3 = x'^2_1 + x'^2_2, d_4 = x_5$. 通过计算可得 $\mathrm{prem}(g, CS) = 0$,因此定理在非退化条件 $d_i \neq 0(i = 1,\cdots,4)$ 下是正确的.如果想进一步知道在退化的情形下,定理是否成立,可以将非退化条件的多项式逐次添加到假设微分多项式组中,并重复上述证明过程,这里不再冗述.

例2(开普勒-牛顿问题) 利用吴方法在微分域上的推广,可以研究开普勒(Kepler)经验定律与牛顿引力定律之间的导出情况.下面通过一简单情形来说明吴方法在这种问题中的应用.

开普勒定律:

($K1$) 行星绕太阳以椭圆轨道运行,并以太阳为其一焦点.

($K2$) 从太阳到行星的向量在相同的时间内扫过相同的面积.

牛顿引力定律:

($N1$) 行星的加速度与太阳到行星的距离的平方成反比.

($N2$) 行星的加速度向量总指向太阳的方向.

下面要证明($N1$) 可以从($K1$) 和($K2$) 导出.

为简单起见,不妨设太阳处于原点(0,0),椭圆的中心位于 x 轴上.假设行星的坐标为$(x(t), y(t))$,其中 t 为时间,则($K1$) 和($K2$) 可表述为

$$h_1 = r^2 - x^2 - y^2 = 0, \quad h_2 = a^2 - x''^2 - y''^2 = 0,$$
$$k_1 = r - p - ex = 0, \quad p' = 0, \quad e' = 0,$$
$$k_2 = y'x - x'y - h = 0, \quad h' = 0,$$

其中 a 为行星的加速度,r 为太阳到行星的距离.

($N1$) 和($N2$) 可以写为

$$n_1 = (ar^2)' = 0, \quad n_2 = x''y - y''x = 0.$$

显然 k_2 和 n_2 是等价的.为方便记,可以假定 $a \neq 0$($a = 0$时,n_1 显然是成立的).取序关系 $p < e < f < x < y < r < a$,则利用分解公式有

$$d\text{-Zero}(k_1, p', e', h_1, h_2, n_2/a) = d\text{-Zero}(\mathrm{ASC}_1/a) \cup d\text{-Zero}(\mathrm{ASC}_2/a),$$

其中 $\mathrm{ASC}_1 = p'$

$$e'$$
$$((e^3 - 1)x^3 + (3c^2 - 1)px^2 + 3epx + p^3)x'' + pxx'$$
$$y^2 - (e^2 - 1)a^2 - 2pex - p^2$$

$$\mathrm{ASC}_2 = \begin{array}{l} r - p - ex \\ a^2 - y''^2 - x''^2 \\ p \\ e' \\ y^2 + (-e^2+1)x^2 \\ r - ex \\ a^2 - y''^2 - x''^2 \end{array}$$

通过计算可得 $\mathrm{prem}(n_1, \mathrm{ASC}_1) = 0$，$\mathrm{prem}(n_1, \mathrm{ASC}_2) \neq 0$，因此"从($K1$)和($K2$)可导出($N1$)"是一般正确的．但不是严格正确的．如果进一步假定 $p \neq 0$，即椭圆不退化成两条直线，并再次利用分解定理，可以得知($N1$)总可以从($K1$)和($K2$)导出．这样就解决了开普勒 - 牛顿问题．实际上，利用零点分解定理可以由开普勒经验公式自动发现牛顿定理([23、20])．

3 构造性实代数几何

实代数几何主要是研究多项式方程在实闭域上定义的零点结构．因为实闭域的结构比复数域要复杂得多，实代数几何的进展一直很缓慢．近年来，由于人们认识到很多高科技问题，特别是机器人与计算机视觉，可以归结为实代数几何问题，兴起了构造性实代数几何的研究．本章将介绍构造性实代数几何的几个主要结果．

3.1 实闭域上的量词消去理论

在20世纪30年代，A. Tarski证明了实闭域的可判定性，从而证明了初等几何与初等代数的可判定性．这里，"初等"是指一阶量词逻辑所能描述的公式．严格定义如下：

变量 $\stackrel{\mathrm{def}}{=}$ | x_1 | x_2 | $\cdots$ | x_r |

多项式 = | 变量 | -1 | 0 | 1 |

| 多项式 + 多项式 | | 多项式 × 多项式 |

原子公式 $\stackrel{\mathrm{def}}{=}$ | 多项式 = 多项式 | 多项式 > 多项式 |

公式 $\stackrel{\mathrm{def}}{=}$ | 原子公式 | ~ 公式 | 公式 V 公式 |

| (变量) 公式 | (V 变量) 公式 |

可以将公式表示为**前束公式**

$$F = (Q_k x_k)\cdots(Q_r x_r)\varphi(x_1, \cdots, x_r), \tag{3-1}$$

其中 Q_i 代表量词，$\varphi(x_1, x_2, \cdots, x_r)$ 为一个无量词公式．在 F 中 $x_1, x_2, \cdots, x_{k-1}$ 被称为自由变量．而 $x_k, x_{k+1}, \cdots, x_r$ 被称为约束变量．仅含约束变量的公式称为语句．假定初等几何或初等代数中的任一命题都对应一个语句．

量词消去法的一般问题：对任一形如(3-1)的公式 F，给出一不含量词的公式

$\varphi(x_1,\cdots,x_{k-1})$ 与之等价.

不难看到,当 F 为一语句时,量词消去法给出该句子的真值.从而证明或否证了对应的命题.

在已有的解决上述问题的算法中G.Collins提出的柱型代数分解算法(CAD)最为有效,并且已在计算机上实现.这里只介绍CAD算法一种简化形式,一般算法可参阅[24]

CAD算法基于下面想法:一个句子的真值仅依赖于多项式的符号 —— 正、负或零,所以若在一区域内公式中多项式的符号不变,则公式的真值也不会变化. CAD算法即将 R^r 分解为有限个柱型区域,使得相应公式中出现的所有多项式在每一区域上不变号.这样,为了证明公式在整个空间上正确与否,只需在每一区域上取一点带入公式验证即可.换句话说,为了说明一类图形的正确性,只需取其中若干个验证即可.

1. CAD的定义

R^r 中非空间的连通开集称为区域.设 S 为 R^{r-1} 中一区域,$(S) = S \times R$ 称为 S 上的柱体. S 上满足条件 $f_1 < f_2 < \cdots < f_k$ 的连续实函数可以将 (S) 分解如下:

$$(S,f_1,\cdots,f_k) = \bigcup_{i=1}^{k}\{(\alpha,b) \mid \alpha \in S, b = f_i(\alpha)\} \bigcup_{i=1}^{k}\{(\alpha,b) \mid \alpha \in S, f_i(\alpha) < b < f_{i+1}(\alpha)\}.$$

(S) 的这一分解称为由 $f_1,f_2,\cdots,f_k$ 确定的栈,其中第一部分称为 (S) 的截面,第二部分称为分解的截段.若 $k = 0$,则 (S) 上的栈即其自身.

R^r 的CAD可归纳定义如下:

1° $r = 0$. $R^0 = \{p\}$ 的CAD即自身.

2° $1 \leq r$.设 $(S_1,S_2,\cdots,S_t)$ 为 R^{r-1} 的一个CAD,对 $i = 1,2,\cdots,t$,设 $f_{i1} < f_{i2} < \cdots < f_{im}$ 为 S_i 上的连续实值代数函数,则 R^r 的CAD可如下给出:$\mathrm{CAD}(R^r) = \bigcup_{i=1}^{t}(S_i,f_{i1},\cdots,f_{im})$.

设 S 为 R^r 的一个CAD, S 中的集合称为该分解的胞腔.下面定义胞腔的指标.对于 R^1,将 R^1 的胞腔从左到右排列每一胞腔的指标即其在这一排列中的序数.设 $C \in S$ 为 $R^r(r > 1)$ 的一胞腔,假定 $C \in (S_i,f_{i1},\cdots,f_{i,r-1})$,其中 S_i 为 R^{r-1} 的一胞腔, S_i 的指标为 $(i_1,i_2,\cdots,i_{r-1})$.又设 C 在栈 $(S_i,\cdots,f_{ir})$ 中自上而下排列为第 i_r 个,则 C 的指数为 $(i_1,\cdots,i_r)$.若 $C \in S$ 的指标为 $(i_1,\cdots,i_r)$,则 $i = \mathrm{par}(I_j)$ 为 C 的维数(其中 $\mathrm{par}(I_j) = 1$,若 I 为奇数;否则 $\mathrm{par}(I_j) = 0$).称 C 为 I- 胞腔,不难看到一个 I- 胞腔同胚于 R^I.

设 $S = (S_1,\cdots,S_\mu)$ 为 R^r 的CAD,β 称为 S 的一个基点,如果 $\beta = (\beta_1,\cdots,\beta_\mu)$,使得 $\beta_i \in S_i(i = 1,\cdots,\mu)$.$\beta$ 称为 S 的柱型代数基点(c,a,s),若下面条件被满足:① 每个 β_i 的坐标为代数数,② 设 S_i、S_j 的指标分别为 $(i_1,i_2,\cdots,i_r)$,$(j_1,j_2,\cdots,j_r)$,则由 $i_k = j_k(k = 1,2,\cdots,p)$ 可推得 $\beta_{i_k} = \beta_{j_k}(k = 1,2,\cdots,p)$.

另一重要概念是不变性,设 S 为一组整系数多项式.说 S 是 A 不变的,若 A 中

所有多项式在 S 的每一胞腔上不变号(或为正,或为负,或为零).

2. 投影

由定义不难看到,CAD 算法明显是递归的,为了构造 R^r 的 CADS 应首先给出 R^{r-1} 的一 CADS′ 并通过在 S' 的胞腔上造栈来给出 R^r 的 CAD,现在就来给出这一递归步骤的理论基础.

定义 1 P 为 $x_1,x_2,\cdots,x_r$ 的多项式,$V(p)$ 且为 F 的实代数族,S 为 R^{r-1} 中的区域.说 P 在 S 上可线性化,如果 $V(p)\cap Z(S)$ 构成 $Z(S)$ 的 K 个不相交截面.直观上看,F 在 S 上可线性化,当且仅当,对 $\forall\alpha\in S$,$P(\alpha,x_r)$ 有固定的不相同的根.这一性质可由子结式来保证.

设 D 为一 U.F.D,$F,G,D[x]$,$S_i(F,G)$ 是 F、G 的 i 子结式.$S_i(F,G)$ 中 x^i 的系统记为 $\mathrm{psc}_i(F,G)$,称为 F、G 的 i 主结式系数.$\mathrm{psc}_i(F,G)$ 有下列性质.

命题 1 $F,G\in D[x]$,有 $\deg(\gcd(A,B))=k$ 当且仅当 $\mathrm{psc}_i(F,G)=0(i=1,\cdots,k-1)$ 并且 $\mathrm{psc}_k(F,G)\neq 0$.

命题 2 $F\in D[x]$.如果 $k=\deg(\gcd(F,F'))$,则 F 有 $m-k$ 个不相同的根($m=\deg(F(x))$).

引入下列记号,设 $F[x]=a_0+a_1x+\cdots+a_rx^r$,则 $\mathrm{ldcf}(F)=a_r$,$\mathrm{Ldt}(F)=a_rx^r$,$\mathrm{red}(F)=F-\mathrm{Ldt}(F)$.对 $k>1$ 定义 $\mathrm{red}(\mathrm{red}^{k-1}(F))=\mathrm{red}^k(F)$.

$\mathrm{RED}(F)=\{\mathrm{red}^k(F)\mid 0\leqslant k\leqslant\deg(F)$,且 $\mathrm{red}^k(F)\neq 0\}$.

$\mathrm{PSC}(F,G)=\{\mathrm{psc}_i(F,G)\mid 0\leqslant i\leqslant\mathrm{mindeg}(F,G)\rfloor\lfloor\mathrm{psc}_i(F,G)\neq 0\}$.

$F\in D[x_1,x_2,\cdots,x_r]$.将 F 看做 x_r 的多项式,记

$$\mathrm{PROJ}_1(F)=\bigcup\{\{\mathrm{ldt}(G)\}\cup\mathrm{psc}(G,G^1),G\in\mathrm{RFD}(F))\}.$$

定理 1 $F\in D[x_1,x_2,\cdots,x_r]$,$S$ 为 R^{r-1} 中一 $\mathrm{PROJ}_1(f)$ 不变区域,则在 S 上可线性化.

定义 2 $A\subset D[x_1,x_2,\cdots,x_r]$ 为多项式有限集,S 为 R^{r-1} 中的区域.A 在 S 上可线性化是指

1° $\forall F\in A$,F 在 S 上可线性化.

2° 对不同的 F、$G\in A$,F、G 在 $Z(S)$ 上产生的截面不相交.

这一要求等价于 F、G 在 S 有固定个数的公共解,或 $\deg(\gcd(F,G))$ 在 S 上为常量.由命题 1 与命题 2,这一点可通过 psc 来保证.设 F、$G\in D[x_1,x_2,\cdots,x_r]$ 令

$$\mathrm{PROJ}_2(F,G)=\bigcup\{\mathrm{psc}(F',G')/F'\in\mathrm{RED}(F),G'\in\mathrm{RED}(G)\}.$$

对 $A\subset I[x_1,x_2,\cdots,x_r]$,进一步定义

$$\mathrm{PROJ}(A)=\{\mathrm{PROJ}_1(G)\cup\mathrm{PROJ}_2(G,F)/G,F\in A,G\neq F\}.$$

定理 2 $A\subset I[x_1,\cdots,x_r]$,$S$ 为 R^{r-1} 中一 PROJ 不变区域,则 A 在 S 上可线性化.

3. CAD 算法

限于篇幅,这里不再给出求 CAD 的算法,可详见[24].下面用一个例子说明 CAD 的形式.

例 1 令 $A=\{y^2-x^3-x^2\}$,$p=y^2-x^3-x^2$,$r=2$.A 的投影集计算如下:

$$\mathrm{ldef}(p)=1,\quad \mathrm{psco}(p,p')=(x^3+x^2)^2,\quad \mathrm{psc}_1(p,p')=1,$$

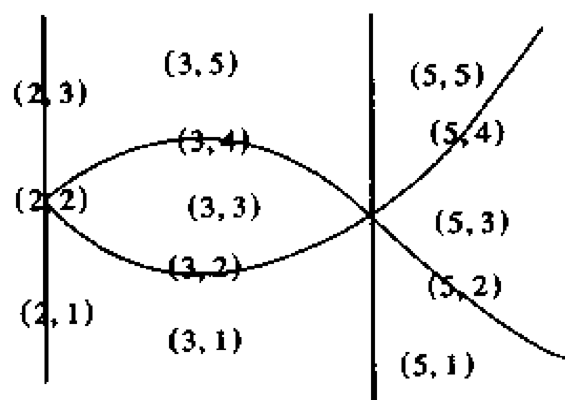


图 3-1

$$\mathrm{Ldcf}(\mathrm{red}(p)) = x^3 + x^2, \mathrm{psco}(\mathrm{red}(p), \mathrm{red}(p'))') = 1.$$

所以只另考虑 $M = \{x, x+1\}$. 其 CAD 可马上给出：

$$S(R^1) = \{(-2)(-1)(-1/2)(0)(1)\}$$

$$I(R^1) = \{(1)(2)(3)(4)(5)\}.$$

现在，比如考虑其中第五个胞腔，对应的多项式为：$A(1,r) = Y^2 - 2$，两个零点为 $\pm\sqrt{2}$. 与之对应的栈为：

$$S(R^2)_5 = \{(1,-2)(1,-\sqrt{2})(1,0)(1,\sqrt{2})(1,2)\}$$

$I(R^2)_5 = \{(5,1)(5,2)(5,3)(5,4)(5,5)\}$，如图 3-1 所示.

3.2　多项式的完全判别系统

大家知道，二次方程 ax^2+bx+c 的判别式 $D = b^2-4ac$ 给出了方程的根的性态. 对于一个 n 次多项式方程 $P(x)=0$，高小山在[25]中定义了 $P(x)=0$ 的复完全判别式，并给出了其计算方法. 特别是首次给出了五次方程的复完全判别式.

设 $t = (n_1, n_2, \cdots, n_k)$ 是 n 的一个拆分，可以求得一 $P(x)$ 的系数的多项式组 D_t，使得如果 $c \in \mathrm{Zero}(D_t)$，则将 c 代入 $P(x)=0$ 中，所得方程得根的分布为 $(n_1, n_2, \cdots, n_k)$，即这一方程有 $n_1, n_2, \cdots, n_k$ 个重根.

如果考虑方程的实根，则需要实根、虚根的重数. 这种情形比复数情形要困难的多. 古典结果已给出四次多项式的根的完全分类. D.S.Arnon 借助 Collins 的 CAD 方法自动给出四次多项式的正定条件.

杨路、侯晓荣和曾振柄在文献[26]中提出了对任意次数的多项式建立完全判别系统的一个新算法，计算出了 5 至 10 次多项式的完全判别式. 应当指出，著名的 Sturm 定理只能用于确定数值系数多项式实根的状况. 直接用 Sturm 定理来建立一个判别系统的尝试是非常没有效率的. 例如，对于文字系数的下列七次多项式，

$$x^7 + px^5 + qx^4 + rx^3 + sx^2 + tx + u$$

在一部 PC 机上(内存 16 兆)计算它的 Sturm 序列，程序(用 MAPLE 编写)运行 1000 多秒后溢出. 用新的算法和程序(仍用 MAPLE 实现)，在同一部机器上产生一个判别式序列仅用 1 秒.

例 2　五次多项式 $g_5 = x^5 + px^3 + qx^2 + r^2 + s$ 的根的分类如下：

(1) $D_5 > 0, D_4 > 0, D_3 > 0, D_2 > 0$: $\{1,1,1,1,1\}$,

(2) $D_5 > 0, D_4 \leqslant 0, D_3 \leqslant 0, D_2 \leqslant 0$: $\{1\}$,

(3) $D_5 < 0$: $\{1,1,1\}$,

(4) $D_5 = 0, D_4 > 0$: $\{2,1,1,1\}$,

(5) $D_5 = 0, D_4 < 0$: $\{2\}$,

(6) $D_5 = 0, D_4 = 0, D_3 > 0, E_2 \neq 0$: {2,2,1},
(7) $D_5 = 0, D_4 = 0, D_3 > 0, E_2 = 0$: {3,1,1},
(8) $D_5 = 0, D_4 = 0, D_3 < 0, E_2 \neq 0$: {1},
(9) $D_5 = 0, D_4 = 0, D_3 < 0, E_2 = 0$: {3},
(10) $D_5 = 0, D_4 = 0, D_3 > 0, D_2 \neq 0, F_2 \neq 0$: {3,2},
(11) $D_5 = 0, D_4 = 0, D_3 > 0, D_2 \neq 0, F_2 = 0$: {4,1},
(12) $D_5 = 0, D_4 = 0, D_3 > 0, D_2 = 0$, : {5},

其中 D_i 是 p、q、r、s 的多项式.这6个以系数 p、q、r、s 为变元的表达式构成一个判别系统,它对上述五次多项式根的分类已经完全够用了.上述分类的右边一栏描述根的分布情况,例如,{1,1,1,1,1} 表示5个单实根,而{2,2,1} 表示两个2重根,而情况(8) 有两个2重虚根.

下面对杨路等提出的方法作一简述:

给了一个一般的 n 次多项式 $f(x)$,其系数自高次至低次为 $a_0, a_1, \cdots, a_n$.将 $f(x)$ 的导数的系数写成 $b_0, b_1, \cdots, b_n$.其中 $b_0 = 0$.这样作的目的是为了使 $f'(x)$ 形式上具有与 $f(x)$ 相同的次数.

将 $f(x)$ 与 $f'(x)$ 的 Sylvester 矩阵叫做 $f(x)$ 的**判别矩阵**,记为 Discr(f).用 D_k 记判别矩阵 Discr(f) 的前 $2k$ 行和前 $2k$ 列所构成的行列式,即判别矩阵的各偶阶主子行列式,$k = 1, 2, \cdots, n$(注意 Discr(f) 是一个 $(2n+1) \times (2n+1)$ 的矩阵).

将 n 个偶阶主子行列式的有序组 $\{D_1, D_2, \cdots, D_n\}$ 叫做多项式 $f(x)$ 的**判别式序列**,并将 [sign(D_1), sign(D_2), sign(D_n)] 叫做**判别式序列的符号表**.

若 $[E, 0, 0, \cdots, 0, F]$ $(E * F \neq 0)$ 是符号表中的一段,将其中全为0的部分代之以 $[-E, -E, E, E, -E, -E, \cdots]$,这样得到的新表,中间不再有0,称之为**符号修订表**.

- 相异的实根和虚根的个数

定理 3　给定一个实系数多项式,如果它的判别式序列的符号修订表的变号数为 u,则 $f(x)$ 有 u 对不同的共轭虚根.又设其符号修订表中非零项的数目为 v,则 $f(x)$ 有 $v - 2u$ 个不同的实根.

- 重因子序列和完全判别系统

令 M 是一个 n 次多项式 $f(x)$ 的判别矩阵

用 M_k 记 M 的 $2k$ 行构成的子矩阵,$k = 1, \cdots, n$,并以 $M(k, i)$ 记由 M_k 的前 $2k-1$ 列及第 $2k+1$ 列构成的子矩阵,$k = 1, 2, \cdots, n$; $i = 0, 1, \cdots, n-k$.然后构作诸多项多式:

$$\Delta_k(f) = \sum_{i=0}^{k} \det(M(n-k, i)) * x^{k-i},$$

这里 $k = 0, 1, \cdots, n-1$.序列 $\Delta_0(f), \Delta_1(f), \cdots, \Delta_{n-1}(f)$ 叫做 $f(x)$ 的**重因子序列**.

引理 1　如果 $f(x)$ 的判别式序列的符号修订表中的0的个数为 k,则

$$\Delta_k(f) = \text{g.c.d}(f(x), f'(x)),$$

这里 g.c.d 表示最大公因子.令 U 表示

$$\Delta_0(f),\Delta_k(f),\Delta_j(\Delta_k(f)),\Delta_i(\Delta_j(\Delta_k(f))),\cdots$$

等多项式集的并,即所有不同层次的重因子序列之并,U 中每个多项式各有一个判别式序列,所有这些判别式序列组成 $f(x)$ 的完全判别系统,记为 D.S(f).

- 重复部分与 Δ 序列

为方便起见,令 $\Delta(f)$ 表示 g.c.d($f(x),f'(x)$),并称之为 $f(x)$ 的重复部分.又令

$$\Delta^0(f)=f,\quad \Delta^i(f)=\Delta(\Delta^{i-1}(f)),\quad i=1,2,\cdots$$

将 $\Delta^0(f),\Delta^1(f),\Delta^2(f),\cdots$ 叫做 $f(x)$ 的 Δ 序列.

为了决定 $f(x)$ 的实根和虚根的数目和重数,无须用到整个的判别式系统 D.S(f);只用 $f(x)$ 的 Δ 序列中那些多项式的判别式序列就足够了.

引理 2　如果 $\Delta^1(f)$ 有 k 个重数为 $n_1,n_2,\cdots,n_k$ 的实根,并且 $\Delta^{i-1}(f)$ 有 m 个不同实根,则 $\Delta^{i-1}(f)$ 有 k 个重数为 $n_1+1,n_2+1,\cdots,n_k+1$ 的实根和 $m-k$ 个单实根.

同样的讨论也适用于虚根.根据上述定理和引理,可得到根的分类的一个完备算法.

3.3　构造性实代数几何的应用

3.3.1　不等式自动证明

构造性实代数几何的一个明显的应用是不等式自动证明.Collins 的 CAD 方法经 Arnon 实现后已经能够证明非平凡的不等式.近来,Hong 的实现在效率上又有所提高.Wesipfenig 等人最近的工作表明,如果把消去理论限于二次方程,则可以得到高效的不等式证明器.

在[27]中,周咸青与高小山用吴零点分解定理证明不等式.这一算法虽然不完全,但却能证明相当困难的约 40 个不等式.在[28]中,将吴零点分解定理与 Collins 的 CAD 方法相结合.首先用吴零点分解定理将问题简化,如果遇到高次方程的不等式问题再调用 Collins 的 CAD 程序.用这一程序对所谓 8_3 几何构型问题做了全面分析.

例 3　(8_3 几何构型问题)问实平面上是否存在 8 条线与 8 个点,各不相同,且每三个点在一条线上,每三条线经过一个点.

这一问题的答案是否定的.但是,如果允许点、线相同,则可以找到所谓退化的 8_3 几何构型.在[28]找到了所有次类退化构型.

最近,杨路等人将完全判别系统与 Collins 的 CAD 方法相结合,针对常见的几何不等式问题开发了 Bottema 程序[29].该程序首次快速证明了大量非平凡不等式,其中还包括很多未解决的问题.Bottema 程序的开发为几何不等式的自动证明画上了圆满的句号.

多项式的正定判定问题是不等式证明的一个特殊问题.Hilbert 十七问题猜测

一个正定多项式总可以表示为若干有理函数的平方和. Artin 在 1929 年证明了这一猜想. 但是, 这一问题的构造性证明还未给出.

3.3.2　全局优化算法

吴文俊在[30]中提出了基于拉格朗日算子与吴零点分解定理的不等式证明方法后, 又进一步将这一方法推广用于解决全局优化问题, 证明了有限核定理.

定理 4　设 D 是欧氏空间中一个矩形闭区域, PS、f 为 D 上多项式组与一个多项式, 且 f 在 PS 的零点上存在极大值或极小值, 则存在一算法可以找到一个 D 的有限子集 K, 使得 f 的极大或极小值都在 K 上达到.

这一算法可以用来解决不等式证明、非线性规划、机器人路径等问题. 王定康用这一方法证明了数十个三角不等式.

3.3.3　路径问题

具体讲, 需要判定一个由等式、不等式定义的区域是否连通. 还可以进一步求最短连通路径. 如果考虑将特定形状的物体从一点搬运到另外一点, 即所谓的 Piano 运动问题. 这些问题在理论上都可以有 CAD 算法解决. 但还不存在高效率算法.

3.3.4　空间定位问题

下面的 PnP 问题是计算机视觉中经典的定位问题. 这一问题在计算机动画、机器人、制图学中都有应用.

***PnP* 问题**　给出了 n 个点 P_i 的相互位置及这些点对于点 P 的视角 P_iPP_j, 求点 P 的位置.

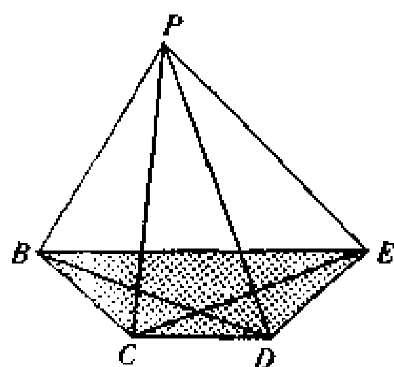


图 3-2

高小山应用吴零点分解定理给出了 $P3P$ 问题的完整解析解, 给出了若干具有几何意义的判定条件[31]. 杨路应用完全判别系统部分给出了 $P3P$ 解的分类情况[32].

4　代数方程组求解算法与应用

代数方程组求解, 是非线性数学中最为基本、最为重要的问题. 相关算法也在许多文章和专著中有所讨论, 如[33]. 鉴于篇幅问题, 本节将只就有关吴方法的代数方程组求解算法及其应用进行讨论.

4.1　代数方程组求解算法

随着吴方法的不断发展及其算法的不断改进, 吴方法的应用已不仅仅局限在

定理的机器证明领域，在诸如理论物理、机构设计、机器人学、计算机辅助几何设计等领域都已获得了成功的应用，特别是吴方法中的整序原理及其零点分解定理为多元多项式方程组的求解开辟了一条新的途径. 这一方法具有以下特点：可以给出方程组解的完整结构；在高维情形可以给出解流形.

设有一组代数方程

$$\mathrm{HS} = \{h_1(x_1,\cdots,x_n) = 0,\cdots,h_r(x_1,\cdots,x_n) = 0\}$$

和一组非等式方程组

$$\mathrm{DS} = \{d_1(x_1,\cdots,x_n) \neq 0,\cdots,d_s(x_1,\cdots,x_n) \neq 0\}.$$

由吴-Ritt零点分解定理得

$$\mathrm{Zero}(\mathrm{PS}/D) = \bigcup_{i=1}^{t}\mathrm{Zero}(\mathrm{AS}_i/DJ_i),$$

其中 $\mathrm{PS} = \{h_1,h_2,\cdots,h_r\}$，$D$ 是多项式 d_i 的乘积；AS_i 是不可约升列；J_i 是 AS_i 中多项式的初式的乘积. 由升列的定义知道，一个升列 $\mathrm{AS} = \{A_1,A_2,\cdots,A_p\}$ 中不同的多项式 A_i 的类各不相同. 假设 $cls(A_i) = k_i$，并将重新命名为 y_i，其余变量命名为 $u_1,u_2,\cdots,u_q$，则 AS 变为如下形式

$$(\mathrm{AS}):\quad \begin{array}{l} A_1(u_1,\cdots,u_q,y_1),\\ A_2(u_1,\cdots,u_q,y_1,y_2),\\ \cdots\cdots\\ A_p(u_1,\cdots,u_q,y_1,\cdots,y_p), \end{array}$$

其中 $u_1,u_2,\cdots,u_q$ 称为AS的参数. 形象地讲，吴方法就是将一个一般的代数方程组变为若干三角形式的方程组. 三角形方程组的结构已经基本清楚. 方程求解的吴方法就是以此为基础进行的. 下面分类说明.

4.1.1 方程求解

如果升列 AS 无参数，则 AS 变为

$$A_1(y_1),A_2(y_1,y_2),\cdots,A_p(y_1,\cdots,y_p).$$

这时AS的零点 $\mathrm{Zero}(AS/I)$ 很容易求得. 因为 $A_1(y_1) = 0$ 是 y_1 的单变量方程，可以用已知方法求其解；然后将 y_1 的解代入 $A_2(y_1,y_2) = 0$ 可以得到一个 y_2 的单变量方程. 依次类推，可以求得 $\mathrm{Zero}(\mathrm{AS}/I)$. 下面举例说明吴方法在方程求解中的应用.

（1）曲面连接问题　在几何设计中有一大类问题可以一般地描述为：给出 $\mathbb{R}^3$ 中的不可约代数曲线 C_i、C_j、C_k，$i\in I$，$j\in J$，$k\in K$，其中 I、J、K 都为有限指标集，以及两组不可约代数曲面，S_j、S_k、$j\in J$，$k\in K$，分别包含曲线 C_j 与 C_k，确定一给定次数 m 的代数曲面 F，使得

1° F 通过所有的 C_i、C_j、C_k.

2° F 沿曲线 C_j 与 C_k 分别与曲面 S_j、S_k 光滑相切.

3° F 沿 C_k 对 S_k 有相同的曲率.

对于这类问题，可以一般地解决如下（[34]）：假设 F 的表达式为

$$F(x,y,z) = \sum_{i,j,k} a_{ijk}x^iy^jz^k,$$

其中 a_{ijk} 为待定系数．由于 C_i、C_j、C_k 都是不可约的代数曲线，因此由 1° 可知 F 对诸曲线余式应为0，故可得一组关于 a_{ijk} 的代数方程组，设为 $PS_1=0$．同样利用代数几何中的有关理论，可以从 2° 和 3° 得到关于 a_{ijk} 的另外两组多项式方程组：$PS_2=0$ 与 $PS_3=0$．然后利用吴方法通过对 $PS_1\cup PS_2\cup PS_3$ 进行整序处理，可得出 F 的精确表达式或 a_{ijk} 应满足的条件，从而达到问题的解决．下面我们来看一个例子：

例 1 设 $S_1=z^2+y^2-r_1^2$，$S_2=z^2+x^2-r_2^2$ 为 $\mathbf{R}^3$ 中的两个圆柱面，$C_1=\{x-d_1,S_1\}$，$C_2=\{y-d_2,S_2\}$ 分别为 S_1 与 S_2 上的曲线（不可约），试确定 3 次代数曲面 F，使得沿曲线 C_1 和 C_2 分别与 S_1 和 S_2 光滑相切．

按照上面所述的思路，这个问题可以化成求解由 28 个多项式方程组成的多项式方程组的问题．将吴方法应用于这些方程组，可得所述三次曲面存在的必要条件为 $r_1^2+d_1^2=r_2^2+d_2^2$．这时唯一可能的曲面为

$$z^2(yd_2+xd_1-d_2^2-d_1^2)+(y^3d_2+x^3d_1)+yx(yd_1+xd_2-2d_2d_1)-(y^2+x^2)(d_2^2+d_1^2)+yd_2(d_1^2-r_1^2)+xd_1(d_2^2-r_2^2)+(r_2^2+d_1^2)(r_2^2+d_2^2)-2d_1^2d_2^2=0.$$

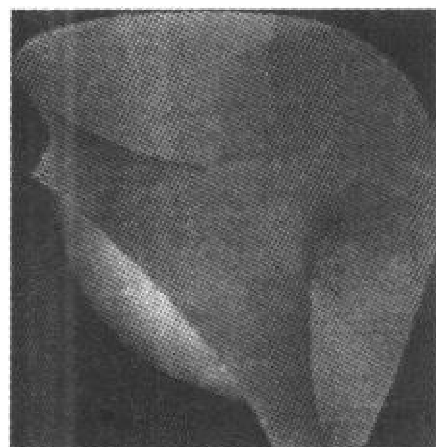


图 4-1

(2) 星体中心构型的确定　设 n 个星体的质量为 $m_1,m_2,\cdots,m_n$．在牛顿引力作用下，这些运动的星体在某时刻的空间位置记为 $r_1,r_2,\cdots,r_n$．这些星体的质量和位置

$$[m_1,\cdots,m_n;r_1,\cdots,r_n]$$

称为这些星体 m_i 在位置 r_i 的一个**构型**．星体的构型称为**中心构型**，如果存在这些星体的一组初速度使得它们在运动中与初始状态保持相似．关于中心构型，有下面猜想．

Wintner 猜想　对于任意质量，只有有限个中心构型．

设 $q_3(n)$、$q_2(n)$、$q_1(n)$ 分别为 n 个给定质量的星体在空间一个平面或一条直线上中心构型的个数．下面是一些已知结果．

$q_1(2)=q_2(2)=q_3(2)=1$；

$q_1(3)=3$；（欧拉，1767）；

$q_2(3)=q_3(3)=4$；（拉格朗日，1772）；

$q_1(n)=n!/2$；（莫尔顿，1910）．

吴文俊在[35]中证明中心构型的决定可以化为代数方程的求解问题．因而，通过吴方法可以求出具体的中心构型的个数．例如考虑平面中心构型问题，由牛顿力学不难推出中心构型的质量重心是保持不变的，可以取这一重心为坐标系的原点，则有

$$m_1r_1+m_2r_2+\cdots+m_nr_n=0.$$

设星体以角速度 ω 绕坐标原点转动,则有下面运动方程

$$\omega^2 y_i = \sum_{j\neq i}\frac{y_{ij}}{r_{ij}^3},\quad i = 1,\cdots,n,$$

$$\omega^2 x_i = \sum_{j\neq i}\frac{x_{ij}}{r_{ij}^3},\quad i = 1,\cdots,n,$$

其中 $x_{ij} = x_i - x_j, y_{ij} = y_i - y_j, r_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2}$. 引进复数坐标:

$$u_i = x_i + \mathrm{i}y_i,\quad v_i = x_i - \mathrm{i}y_i,\quad i = 1,\cdots,n;\quad \mathrm{i} = \sqrt{-1};$$

$$z_{ij} = \frac{1}{(u_i - u_j)r_{ij}},\quad \omega_{ij} = \frac{1}{(v_i - v_j)r_{ij}},\quad i,j = 1,\cdots,n,\ i\neq j;$$

$$m_0 = -\omega^2.$$

原来的运动方程将变为下面方程组:

$$m_0 u_i + \sum_{j\neq i} m_j\omega_{ij} = 0,\quad i = 1,\cdots,n.$$

$$m_0 v_i + \sum_{j\neq i} m_j z_{ij} = 0,\quad i = 1,\cdots,n.$$

$$(u_i - u_j)z_{ij}r_{ij} - r^3 = 0,$$

$$(v_i - v_j)\omega_{ij}r_{ij} - r^3 = 0,$$

$$r_{ij}^2 - (u_i - u_j)(v_i - v_j) = 0,$$

$$r_{ij} - r_{ji} = 0,\ z_{ij} - z_{ji} = 0,\ \omega_{ij} - \omega_{ji} = 0,\ i,j = 1,\cdots,n, i\neq j.$$

上面方程组的解如果满足下面"现实性"条件,则相应的星体是一个中心构型. 所谓现实性条件,即 $m_0, m_1,\cdots,m_n$ 是实数,且 $m_0 < 0, m_i > 0$, r_{ij} 为正实数, u_i、v_i 是共轭复数, $\mathrm{i} = \sqrt{-1}$, z_{ij}、ω_{ij} 非零.

现在考虑平面三体中心构型. 也就是说将质量 m_1、m_2、m_3 看成参数,求解变量 u_i、v_i、m_0、r_{ij}. 为了解方程方便,用下面新变量代替变量 u_2、u_3、v_1、v_2:

$$u_{12} = u_1 - u_2,\quad u_{13} = u_1 - u_3,\quad v_{12} = v_1 - v_2,\quad v_{13} = v_1 - v_3.$$

则求平面三体中心构型的运动方程等价于求下面方程满足现实性条件的解:

$$q_1 = u_1(m_1 + m_2 + m_3) - u_{13}m_3 - u_{12}m_2,$$

$$q_2 = u_{13}^2 r_{23} r_{13}(r_{12}^3 m_0 + m_2 + m_1) - v_{13}v_{12}r_{13}(r_{12}^3 m_0 + m_2 + m_1) - v_{13}v_{12}r_{12}m_3(r_{23} - r_{13}) - v_{12}^2 r_{23} r_{12} m_2,$$

$$q_3 = u_{13}^2 r_{23} r_{13} m_2 + v_{13}v_{12}r_{23}r_{13}(r_{13}^2 r_{12} m_0 - m_2) + v_{13}v_{12}r_{12}(r_{23}m_3 + r_{23}m_1 + r_{13}m_1) - v_{12}^2 r_{22} r_{12}(r_{13}^3 m_0 + m_3 + m_1),$$

$$q_4 = v_1 u_{13} u_{12} u_{12} r_{13} r_{12} m_0 + u_{13} r_{13} m_2 + u_{12} r_{12} m_3,$$

$$q_5 = u_{13}^2 r_{23} r_{13}(r_{12}^3 m_0 + m_2 + m_1) - u_{12}u_{12}r_{23}r_{13}(r_{12}^3 m_0 + m_2 + m_1) + u_{13}u_{12}r_{12}m_3(r_{23} - r_{13}) - u_{12}^2 r_{23} r_{12} m_3,$$

$$q_6 = u_{13}^2 r_{23} r_{13} m_2 + u_{13}u_{12}r_{23}r_{13}(r_{13}^2 r_{12} m_0 - m_2) + u_{13}u_{12}r_{12}(r_{23}m_3 + r_{23}m_1 + r_{13}m_2) - u_{12}^2 r_{23} r_{12}(r_{13}^3 m_0 + m_3 + m_1),$$

$$q_7 = v_{12}u_{12} - r_{12}^2,$$

$$q_8 = v_{13}u_{12} - r_{13}^2,$$
$$q_9 = (v_{13} - v_{12})(u_{13} - u_{12}) - r_{23}^2.$$

应用吴方法可以求得这一方程的所有已知解[35].在[36]中应用相似的方法还求得了关于四体中心构型的新结果.

4.1.2 通过预解式求解方程

在上面求解方程的过程中,需要连续求解若干非线性方程.这里会有计算误差的连续积累问题,但可以通过预解式来避免这一问题.设 AS 是一个形如(AS)的零维升列.引进新变元 $w = c_1y_1 + \cdots + c_ny_n$,其中 c_i 是随机选取的整数,于是有([10])

定理 1 若随机选取 c_i,并在变元顺序 $u < w < y_1 < \cdots < y_p$ 下重新使用整序算法

$$\mathrm{Zero}(\mathrm{AS} \cup \{w - c_1y_1 - \cdots - c_ny_n\}/J = \mathrm{Zero}(\mathrm{AS}'/J)$$

其中 J 是 AS 的初式的乘积.则 AS′ 具有下面形式的概率为 1.

$$R(w), R_1 = y_1 - U_1(w), \cdots, R_n = y_n - U_n(w),$$

其中 R 称为 AS 的预解式.设 $R(w)$ 的次数为 N,则可以假定 U_i 的次数小于 N.

通过上面定理,可将求解升列 AS 零点的问题化为求解 AS 的预解式中新变元 w 的问题.w 解出后,其余变量可以如下求得:$y_i = U_i(w), i = 1,2,\cdots,n$.下面定理给出了 AS 的解的完全分离.

定理 2 沿用定理 1 中的记号.设 $e_1, e_2, \cdots, e_s$ 为 $R(w) = 0$ 的 $d\dfrac{m^N}{\sqrt{n}(M+2m)^{N+1}}(0 < d < 1)$ 有理近似解,则 $(U_1(e_i), \cdots, U_n(e_i)), i = 1,2,\cdots,n$,为 AS 的 d 近似解.m 与 M 是 $R(w)$ 系数的最小与最大绝对值.

定理 3 沿用定理 2 中的记号.设 $e_1, e_2, \cdots, e_s$ 为 $R(w) = 0$ 的 $S\dfrac{m^N}{8\sqrt{n}(M+2m)^{N+1}}$ 有理近似解,其中 $S = \sqrt{3}2^{\frac{1-N}{2}}N^{-N}T^{1-N}$,$T$ 是 U_i 的系数的最大绝对值,则以 $(U_1(e_i), \cdots, U_n(e_i)), i = 1,2,\cdots,n$,为球心以 $S/8$ 为半径的球中含有 AS 的唯一解.

4.1.3 方程组的流形解与几何公式的自动推导

如果升列 AS 确有参数存在(见公式 4-1)),则实际上是求得了原问题的流形解:$A_1(u_1, u_2, \cdots, u_q, y_1) = 0$.在几何中这种问题经常遇到,例如,几何公式的推导与轨迹方程的计算就是这样两类问题.详见 5.2 节.

流形解的重要应用之一是连杆机构的运动分析.运动分析有两类:已知连杆机构的构成,求该机构上某一点的轨迹及该点的位置与连杆机构的关系.这类问题称为机械设计中的**正解问题**.反过来,求解连杆机构的参数使得连杆机构上一点恰好位于空间的指定位置的问题称为机械设计中的**反解问题**([37]).这两类问题都可以看成公式推导问题.下面给出一个还未彻底解决的求正解的问题.

例 2(Stewart 平台) 如图 4-2 所示,P_1、P_2、P_3、P_4、P_5、P_6 是空间中六个固定

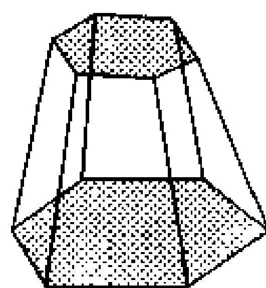


图 4-2

点. Q_1、Q_2、Q_3、Q_4、Q_5、Q_6六个点在一个刚体平台上. P_1Q_1、P_2Q_2、P_3Q_3、P_4Q_4、P_5Q_5、P_6Q_6六根连杆长度可变. 求这六根连杆的长度变化时平台上一点的轨迹.

这是一个非常困难的正解问题，现在还没有完全解决. 最好的结果是：这一轨迹是一个次数为40的多项式所确定的空间图形([38、39]). Stewart平台的研究具有很强的应用背景. 最近研制的基于Stewart平台的数控机床被称为是21世纪的机床.

4.2 杨振宁 - 柏克斯特方程与量子群

杨振宁 - 柏克斯特(B.J.Baxter)方程首先是由杨振宁先生在1967年建立的. 它反映了物理量之间的交换关系. 1982年柏克斯特在研究统计物理模型时使用了同样的方程. 经十多年的深入研究，逐渐发掘出这一方程深刻的数学背景和物理背景. 现在它已是许多数学领域，如低维拓扑、打结理论、辫子群、量子群的基本方程，在理论物理的许多分支中，如统计物理、可积分系统等扮演着关键的角色. 所以，寻找杨振宁 - 柏克斯特方程的解具有重要的理论意义和实际背景.

杨振宁 - 柏克斯特方程(以下简称为YBE)是复数域上的一组代数方程，因此，从理论上讲，应用吴消元法可求出方程的全部解. 但是，一般 n 维情形下这组方程极为复杂，就是在二维情形，这组方程仍然是相当复杂的多项式的代数方程组.

设矩阵 $\boldsymbol{R}$ 和置换矩阵 $\boldsymbol{P}$ 为

$$\boldsymbol{R}=\begin{bmatrix} x_{15} & x_3 & x_1 & x_{11} \\ x_5 & x_{14} & x_{10} & x_2 \\ x_7 & x_9 & x_{13} & x_4 \\ x_{12} & x_8 & x_6 & x_{16} \end{bmatrix}, \qquad \boldsymbol{P}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中 x_i, $i=1,2,\cdots,16$ 为复变元.

命矩阵

$$\boldsymbol{R}_{12}=\boldsymbol{RP}\otimes \boldsymbol{I},\boldsymbol{R}_{23}=\boldsymbol{I}\otimes \boldsymbol{RP},$$

其中

$$\boldsymbol{R}_{12}=\begin{bmatrix} x_{15} & 0 & x_1 & 0 & x_3 & 0 & x_{11} & 0 \\ 0 & x_{15} & 0 & x_1 & 0 & x_3 & 0 & x_{11} \\ x_5 & 0 & x_{10} & 0 & x_{14} & 0 & x_2 & 0 \\ 0 & x_5 & 0 & x_{10} & 0 & x_{14} & 0 & x_2 \\ x_7 & 0 & x_{13} & 0 & x_9 & 0 & x_4 & 0 \\ 0 & x_7 & 0 & x_{13} & 0 & x_9 & 0 & x_4 \\ x_{12} & 0 & x_6 & 0 & x_8 & 0 & x_{16} & 0 \\ 0 & x_{12} & 0 & x_6 & 0 & x_8 & 0 & x_{16} \end{bmatrix}$$

$$
R_{23} = \begin{bmatrix}
x_{15} & x_1 & x_3 & x_{11} & 0 & 0 & 0 & 0 \\
x_5 & x_{10} & x_{14} & x_2 & 0 & 0 & 0 & 0 \\
x_7 & x_{13} & x_9 & x_4 & 0 & 0 & 0 & 0 \\
x_{12} & x_6 & x_8 & x_{16} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x_{15} & x_1 & x_3 & x_{11} \\
0 & 0 & 0 & 0 & x_5 & x_{10} & x_{14} & x_2 \\
0 & 0 & 0 & 0 & x_7 & x_{13} & x_9 & x_4 \\
0 & 0 & 0 & 0 & x_{12} & x_6 & x_8 & x_{16}
\end{bmatrix}
$$

则二维杨振宁-柏克斯特方程是

$$
\boldsymbol{M} = [m_{ij}] = \boldsymbol{R}_{12}\boldsymbol{R}_{23}\boldsymbol{R}_{12} - \boldsymbol{R}_{23}\boldsymbol{R}_{12}\boldsymbol{R}_{23} = 0.
$$

由此可以看出，方程组 $m_{ij} = 0, i, j = 1, 2, \cdots, 8$ 由 64 个三次多项式方程组成，包括 16 个未知数，是一组超定方程. 可以料想，这一方程组的求解是非常困难的. 应用吴消元法、采用人机对话的方式，运用多种技巧，成功地求出了二维杨振宁 - 柏克斯特方程的全部解. 这一成功的经历，表明了吴消元法的优越性. 同时似乎也告诉人们，一组数学方程，不管表现的形式如何复杂，只要其具有深刻的数学背景和物理背景，那么这组方程将有丰富的内部结构，只要使用的工具有力，研究方法得当，那么总能对其有所认识. 参阅文献[40].

相应于杨振宁 - 柏克斯特方程的解，可得到量子群. 然而在获得杨振宁 - 柏克斯特方程的解之后，如何具体计算对应的量子群并非易事. 应用代数簇的母元概念，文献[40,41] 给出依据杨振宁 - 柏克斯特方程的解直接计算相应量子群的机械化方法.

4.3　微分系统稳定性与极限环的个数

大家知道，多项式微分系统极限环个数问题，即希尔伯特第 16 问题，是微分方程定性理论研究中困难而又头等重要的问题. 对于平面二次微分系统的极限环个数研究，我国数学家作出过重要贡献. 这一问题目前通行的研究方法是构造微分系统的李雅普诺夫（Liapunov）函数，计算相应的判定量而获得结论. 这一研究过程涉及大量繁杂多项式的推导、简约，这些多变元多项式具有数百项甚至上千项，其计算繁复而冗长. 研究者往往要集中精力、细致耐心地连续工作数月才能取得一些结果. 如果计算中稍有疏漏，将导致结果的谬误. 比之二次系统，三次微分系统的极限环个数研究，其困难程度大幅度增长，有关多项式的推导演算的复杂度以指数形式增加. 因此，仅凭人力的手工作业，关于平面三次微分系统的极限环个数研究将是难以实现的.

吴消元法的建立与发展，为这一问题的研究提供了强有力的数学工具和机械化的研究手段. 以吴消元法为理论基础，综合利用计算机代数中的各类技巧，王东明等开展了这一方向的研究，取得了出乎人们意料的进展（参阅文献[36]）. 他在后续工作中进一步扩展了所取得的成绩，获得一系列重要成果.

众所周知,中心焦点型判定量的性态与极限环的个数紧紧相关,在临界情形最有可能跳出极限环.因此,微分动力系统在原点附近极限环个数研究,主要难点归结为中心焦点判定问题.顺便叙及这一问题和微分系统的稳定性研究有着密切联系.

给定平面 n 次多项式微分系统

$$\begin{cases}\dfrac{\mathrm{d}x}{\mathrm{d}t} = y + P_2(x,y) + P_3(x,y) + \cdots + P_n(x,y),\\[2mm] \dfrac{\mathrm{d}y}{\mathrm{d}t} = x + Q_2(x,y) + Q_3(x,y) + \cdots + Q_n(x,y),\end{cases} \tag{4-1}$$

其中 P_i、Q_i,$i = 2,3,\cdots,r$,是变元 x、y 的 i 次齐次多项式.记 P_i、Q_i,$i = 2,3,\cdots,r$,的系数为 $u_1,u_2,\cdots,u_d$.计算判定量要构造微分系统(4-1)的李雅普诺夫函数.命 $F(x,y) = \sum_{j=2}^{\infty} F_j(x,y)$,其中 $F_j(x,y)$ 是 x、y 的 j 次齐式:

$$F_2(x,y) = \frac{1}{2}(x^2 + y^2),\quad F_j(x,y) = \sum_{k=0}^{j} f_{jk}x^{j-k}y^k,\quad j > 2.$$

沿微分系统(4-1)的积分曲线对 $F(x,y)$ 求导.为使 $\dfrac{\mathrm{d}F}{\mathrm{d}t}$ 在原点的邻域中定号,假设其具有形式

$$\frac{\mathrm{d}F(x,y)}{\mathrm{d}t} = v_3y^4 + v_5y^6 + \cdots + v_{2k-1}y^{2k} + \cdots, \tag{4-2}$$

其中 v_{2k+1},$k = 1,2,\cdots$ 为待定系数,称为李雅普诺夫常数.中心焦点判定问题就是要具体确定这些常数,即对给定的方程(4-1)中多项式 $P(x,y)$、$Q(x,y)$,具体计算出 v_{2k+1},$k = 1,2,\cdots$,是 $P(x,y)$,$Q(x,y)$ 的系数 $u_1,u_2,\cdots,u_d$ 的多项式.由于(4-2)式的左端可写为 $\dfrac{\mathrm{d}F(x,y)}{\mathrm{d}t} = \sum_{j=3}^{\infty} I_j(x,y)$,其中 $I_j(x,y)$ 是 x、y 的 j 次齐式,即

$$I_j(x,y) = \sum_{k=2}^{j}(j-k)f_{jk}x^{j-k-1}y^{k+1} - \sum_{k=1}^{j}kf_{jk}x^{j-k+1}y^{k-1} + \sum_{k=0}^{j}G_{jk}x^{j-k}y^k,$$

其中 $G_{j0},G_{j1},\cdots,G_{jj}$ 都是 f_{lk},$2 \le l \le j-1$,$0 \le k \le l$ 以及 $u_1,u_2,\cdots,u_d$ 的多项式.比较(4-2)式两端的系数则有

$$I_j(x,y) = \begin{cases}0, & j\text{ 为奇数},\\ v_{j-1}y^j, & j\text{ 为偶数}.\end{cases}$$

由此得到一系列关于 f_{jk} 的递推线性方程组.不难证明,从这些方程组可具体解出 f_{jk} 作为 $u_1,u_2,\cdots,u_d$ 多项式,这样就保证了函数 $F(x,y)$ 的具体构造,同时保证了 v_{2k+1},$k = 1,2,\cdots$,也是 $u_1,u_2,\cdots,u_d$ 的多项式.

如果 $v_{2k+1} = 0$,$k = 1,2,\cdots$,则称原点是 n 次系统(4-1)的中心.其它情形则称原点为系统(4-1)的焦点.如果 $v_3 = v_5 = \cdots = v_{2k-1} = 0$ 而 $v_{2k+1} \neq 0$,则称原点为系统(4-1)的 m 重焦点.为判定原点是否是中心,或者当原点是焦点时判定其重数,自然要讨论下述问题:

问题 1　如何决定在条件 $v_3 = v_5 = \cdots = v_{2j-1} = 0$ 之下,v_{2j+1} 是否恒为零.

问题 2 如何利用 $v_3 = v_5 = \cdots = v_{2k-1} = 0$ 简化 v_{2k+1} 的表述形式.

问题 3 寻求一组等价于 $v_3 = v_5 = \cdots = v_{2N+1} = 0$ 的更为简单的条件,使得问题的结果得到简单明确的叙述.

问题 4 构造出具体的微分系统,即具体给定多项式 P_i、Q_i, $i = 2,3,\cdots,n$ 的系数 $u_1,u_2,\cdots,u_d$,使之 $v_3 = v_5 = \cdots = v_{2k-1} = 0$,但 $v_{2k+1} \neq 0$.

可以料想,这些问题的处理绝非易事.为此,王东明编制了专用的计算机计算程序.这一程序的数学原理综合使用了多项式组特征列的计算、多项式对特征列的求余、代数簇的分解、未知关系的自动推导等有关内容,并利用计算机代数的若干技巧,有时还辅之以 Groebner 基的基本性质.例如,只要输入给定的多项式微分系统(即输入适当给定的多项式系数),计算机即可依照给定的指令,输出所需要的多项式 $v_3,v_5,\cdots,v_{2k-1}$.有了这样的计算程序,即可快速准确地计算具体给出的微分系统的判定量,从而利用计算机开展微分系统极限环个数的机械化研究.计算机的高速运算,使得大量繁杂的计算推导可在几分钟(甚至几秒)内即可获得结果,而且保证其准确无误.比如,对具体给定的平面二次微分系统,判定量的计算若用人力的手工作业要耗费大量的精力和宝贵的时间,若利用计算机,由于已编制好专用程序,平面二次系统判定量的计算所需时间将是微不足道的.这样就为数学家的研究工作带来极大的便利.可在更大范围内展开研究工作.正是借助于这种机械化研究手段,在极为困难的平面三次微分系统的极限环研究中取得了一系列成果.鉴于篇幅有限,详情请参看[36].

4.4 一类发展方程的行波解

自从斯考特-鲁塞尔(Scott-Russell)在1834年观察到浅水波中的孤立子现象以来,非线性发展方程行波解(孤立波解,孤立子解)的研究一直为数学家、物理学家所关注.非常有趣的是,正是由于电子计算机的问世和发展,才促使这一领域的研究取得巨大的进展.因此,在开展数学机械化研究时,自然应当关注这一重要的研究领域,力求发挥数学机械化的优势,在某些主要的研究方向取得有意义的成绩.

非线性发展方程具有深刻的物理学、力学背景,为数学界所瞩目.许多大数学家都在这个领域从事过研究,作出了巨大贡献,产生了一些著名的理论和方法,如反散射理论、反散射变换法、完全可积哈密顿系统、代数几何方法、李代数求可积系统的方法等等.近二三十年来,有关的学术专著连续出版,学术论文大量发表,成为一个十分活跃的数学领域.

寻求非线性发展方程的行波解(孤立波解)是一个重要的研究方向.由于非线性发展方程种类繁多,而对每个方程具体找到行波解并不容易,常用的手段是发现或求出某种变换,使得方程变为某种经典的或者较为容易求解的形式,然后再进行求解.但是,发现和构造这些变换并非易事,往往需要新奇的想法,巧妙的构思才能成功.每一个方程用一种特殊的方法,难以发现其规律.

然而,随着具体求得的非线性发展方程的行波解(孤立波解、孤立子解)日益增多,实践的积累使人们发现,一大类非线性发展方程的行波解都是用双曲函数表

示的.这就启示人们,在这类特定形式的、可用双曲函数表示的范围之内去求非线性发展方程的行波解.

这样一来,所求的解的形式已经给定,只是其中的某些参数有待确定.将这种给定形式的行波解代入所考虑的非线性发展方程,经过比较相应的系数,最后可归结为求解这些待定系数所满足的多项式方程组.于是应用吴文俊消元法,可完整求出这种特定形式的行波解,不但重新得到原有的解,而且可以发现大量新的解.这种求非线性发展方程行波解的方法是机械化的,一大类非线性发展方程都可利用这种方法统一地按部就班地求得其行波解.

在前人工作的基础上,文献[42]提出了统一求出一大串非线性发展方程的行波解的机械化方法,文[43]又把这种机械化方法应用到研究非线性发展方程的孤立波解.

在反应扩散方程行波解的研究中,贝洛索夫-扎波琴斯基(B-Z)模型的研讨有重要理论意义和实际背景.本节以其为例,说明上述的机械化方法是如何具体实现的.简化的二维 B-Z 方程是

$$\begin{cases} u_t = \mathrm{d}u_{xx} + u(1 - u - rv), \\ v_t = v_{xx} - suv, \end{cases} \tag{4-3}$$

其中 r、s 为正的参数,d 为耗散常数,未知函数 $u = u(x,t)$、$v = v(x,t)$.

在边条件

$$u(-\infty,t) = 0,\quad v(-\infty,t) = 1,\quad u(+\infty,t) = 1,\quad v(+\infty,t) = 0 \tag{4-4}$$

的限制下,假设未知函数 $u(x,t)$、$v(x,t)$ 具有如下形式:

$$\begin{cases} u = \dfrac{f}{h^2},\quad f = ae^z + e^{2z},\quad h = 1 + e^z, \\ v = \dfrac{g}{h^2},\quad g = 1 + be^z,\quad h = 1 + e^z, \end{cases} \tag{4-5}$$

其中 $z = k(x + ct)$,$k > 0$,$c > 0$,a、b、k、c 为待定系数.

对(4-5)式给出的 $u(x+ct)$、$v(x+ct)$ 进行求导并比较 e^z 的不同幂次的系数,得到待定系数 a、b、c、k 以及 d、s、r 的代数方程组

$$\begin{cases} a(1 - ck + dk^2 - r) = 0, \\ -1 - 2a + a^2 + 2ck - 4dk^2 + 4adk^2 + r + abr = 0, \\ 2 - a - 2ck + ack - 2dk^2 + adk^2 - br = 0, \\ 2ck - bck - 2k^2 + bk^2 - as = 0, \\ -2ck - 4k^2 + 4bk^2 + s + abs = 0, \\ b(ck + k^2 - s) = 0. \end{cases} \tag{4-6}$$

于是,求方程(4-3)的行波解就转化为求解多项式方程组(4-6).应用吴消元法在计算机上具体进行计算,可迅速求出方程组(4-6)的全部解.具有物理意义的解共有下面的六组:

$$a=0,b=0,k=\frac{1}{\sqrt{1+d}},c=\frac{1}{\sqrt{1+d}};$$

$$a=0,b=1,k=\frac{1}{\sqrt{6d}},c=\frac{1}{\sqrt{6d}};$$

$$a=0,b=2,k=\sqrt{\frac{s}{6}},c=5\sqrt{\frac{s}{6}};$$

$$a=\frac{1-3d}{d-2},b=0,k=\frac{1}{2}\sqrt{\frac{4d-3}{d(d-2)}},c=\frac{5}{2}\sqrt{\frac{d}{(d-2)(4d-3)}};$$

$$a=\frac{3d-1}{1-d},b=1,k=\frac{\sqrt{1-2d}}{\sqrt{3(d-d^2)}},c=\frac{d\sqrt{1-2d}}{\sqrt{3(d-d^2)}};$$

$$a=\frac{5-3d}{d-1},b=\frac{7d-9}{3-d},$$

$$k=\sqrt{\frac{(2d-3)(5-3d)}{d(d^2-1)}},\quad c=\sqrt{\frac{d(5-3d)}{(2d-3)(d^2-1)}}.$$

根据吴消元法的零点分解定理,保证了这六组解的完整性.在求解多项式方程组(4-6)的过程中,相应于上述六组解,同时可以得到对参数 d、r、s 所满足的约束条件:

$$\begin{aligned}
&(1+d)r=5d-1, && (1+d)s=6,\\
&3dr=5d-1, && 3ds=1,\\
&r+s=1, && d=1,\\
&2(d-2)r=4d-3, && 2ds=3,\\
&3(1-d)r=3-5d, && 3ds=1,\\
&(d^2-1)r=(3-d)(5d-7), && (d+d^2)s=3(5-3d).
\end{aligned}$$

现在可以写出方程(4-3)的行波解的六组明确表达式如下:

(1) 如果 $d>\frac{1}{5}$,并且 d、r、s 满足第一组条件,则有

$$u(x+ct)=\frac{1}{4}+\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{1}{4}\tanh^2\left(\frac{k(x+ct)}{2}\right),$$

$$v(x+ct)=\frac{1}{4}-\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{1}{4}\tanh^2\left(\frac{k(x+ct)}{2}\right).$$

(2) 如果 $d>\frac{1}{5}$,并且 d、r、s 满足第二组条件,则有

$$u(x+ct)=\frac{1}{4}+\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{1}{4}\tanh^2\left(\frac{(k(x+ct)}{2}\right),$$

$$v(x+ct)=\frac{1}{4}-\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right).$$

(3) 如果 $d=1$,并且 d、r、s 满足第三组条件,则有

$$u(x+ct)=\frac{1}{4}+\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{1}{4}\tanh^2\left(\frac{k(x+ct)}{2}\right),$$

$$v(x+ct)=\frac{1}{4}-\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)-\frac{1}{4}\tanh^2\left(\frac{k(x+ct)}{2}\right).$$

(4) 如果$\frac{1}{4}<d<\frac{3}{4}$,并且 d、r、s 第四组条件,则有

$$u(x+ct)=\frac{1+2d}{4(2-d)}+\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{3-4d}{4(2-d)}\tanh^2\left(\frac{k(x+ct)}{2}\right),$$

$$v(x+ct)=\frac{1}{4}-\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{1}{4}\tanh^2\left(\frac{k(x+ct)}{2}\right).$$

(5) 如果$\frac{1}{3}<d<\frac{1}{2}$,并且 d、r、s 第五组条件,则有

$$u(x+ct)=\frac{2d}{4(1-d)}+\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{1-2d}{2(1-d)}\tanh^2\left(\frac{k(x+ct)}{2}\right),$$

$$v(x+ct)=\frac{1}{4}-\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right).$$

(6) 如果$\frac{3}{2}<d<\frac{5}{3}$,并且 d、r、s 满足第六组条件,则有

$$u(x+ct)=\frac{2-d}{2(d-1)}+\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{2d-3}{2(d-1)}\tanh^2\left(\frac{k(x+ct)}{2}\right),$$

$$v(x+ct)=\frac{3(d-1)}{2(3-d)}-\frac{1}{2}\tanh\left(\frac{k(x+ct)}{2}\right)+\frac{3-2d}{3-d}\tanh^2\left(\frac{k(x+ct)}{2}\right).$$

这样就证明了方程(4-3)在某些条件限制下,行波解有且只有上述六种.在求解过程中还发现,如果适当放松限制,还可求出其它有意义的解.

5　几何自动推理的代数方法

定理证明是最典型的推理.定理证明的机械化思想由来已久,一些原始的想法可以追溯到 17 世纪的 G.W.Leibniz 和 R.Descartes.就是要把一类数学问题当作一个整体,加以考虑,建立一种统一的,确定的证明过程,使得该类中的每个定理,只要按规定的程序步骤机械、刻板地一步一步地实施下去,经过有限步之后,即可判断出定理的正确性.自动推理理论可以分为三类:以 Herbrand 理论及 Resolution 为代表的逻辑方法;以 Newell,Simon 等人为代表的人工智能方法;以 Tarski 理论与吴方法为代表的代数方法.本章主要介绍代数方法,因为这类方法就证明几何定理而言是最有效的.

5.1　几何定理机器证明的吴方法

几何定理证明的机械化可以追溯到 20 世纪初的希尔伯特.代数方法的实质在于把通常数学证明中所固有的质的困难性,代之以计算量的复杂性.70 年代,吴文俊教授提出了"吴方法"之后,使得几何定理的"机器证明"真正成为可能.下面我们简要介绍一下吴方法.详见[4、1、44、45、46].

5.1.1　形式 I

利用吴方法证明几何定理可以分为如下几个步骤：

步 1　几何问题的代数化.建立坐标系，并对定理中的点选取适当的坐标，则定理的假设可以写成多项式方程组 HS：

$$(\mathrm{HS})\quad\begin{cases}h_1(x_1,x_2,\cdots,x_n)=0,\\ \cdots\cdots\\ h_r(x_1,x_2,\cdots,x_n)=0.\end{cases}$$

结论写为 $C(x_1,x_2,\cdots,x_n)=0$.由于任一组满足 HS 的坐标 $x_1,x_2,\cdots,x_n$ 都对应了满足定理假设的一组点，因此定理的证明转化成了判定(HS)的一组解是否也是结论方程 $C(x_1,x_2,\cdots,x_n)=0$ 的一组解的问题.

步 2　整序　令 $\mathrm{PS}=\{h_1,h_2,\cdots,h_n\}$.对 PS 中所出现的变元选取适当的次序，并对 PS 进行整序，求出 PS 的特征到 CS.这时将有 $\mathrm{Zero(PS)}=\mathrm{Zero(CS/I)}\cup\mathrm{Zero(PS}\cup\{I_i\})$，其中 I_i 是 CS 中多项式的初式，I 是所有这些初式构成的集合.

步 3　将结论多项式 $C(x_1,x_2,\cdots,x_n)$ 对 CS 中的多项式按类的高低从大到小依次进行约化，求出 $C(x_1,x_2,\cdots,x_n)$ 对 CS 的余式 $R=\mathrm{prem}(C,\mathrm{CS})$.

如果 $R=0$，则说明 $\mathrm{Zero(CS/}I)\subset\mathrm{Zero}(C)$，即在诸 $I_i\neq0$ 的条件下，HS 的解都是结论方程 $C(x_1,x_2,\cdots,x_n)=0$ 的解，从而说明在诸 $I_i\neq0$ 的条件下，定理是正确的.

步 4　将每一 I_i 分别添入 PS 中，重复进行步 2、步 3，进一步验证初式 $I_i=0$ 的情况下，定理的正确性.

以上证明过程的最后，可能出现的条件 $I_n\neq0$ 称为定理的**非退化条件**.一般来说，一个几何定理的成立，并不是无条件的，即在一些退化的情形下，定理将失去其正确性.通常的几何定理的证明过程中，对这些退化情况的处理是模糊而不明确的，甚至是不加以考虑的，因而严格来说也是不够严密的，事实上也是不可能达到完全严密.用吴方法来证明定理，不仅可以判断出定理的正确性与否，而且还可以在证明过程中，自动找出这些定理赖以成立的非退化条件.

在用吴方法证明定理时，对一些特殊的定理，有时会要用到零点分解定理.这里不再详述，下面来看几个例子：

例 1(Simson 定理)　设 D 是三角形 ABC 的外接圆上任意一点，由 D 向三角形的三边作垂线分别交 BC、CA、AB 于 E、F 和 G，证明 E、F、G 共线.

证明　首先取点的坐标：$A:(0,0)$，$B:(x_1,0)$，$C:(x_2,x_3)$，$O:(x_4,x_5)$，$D:(x_6,x_7)$，$G:(x_8,x_9)$，$F:(x_{10},x_{11})$，$E:(x_{12},x_{13})$.则几何命题中的假设与结论就可以转换为如下代数方程.

命题假设：

$OA=OB$　　$2x_1x_4-x_1^2=0$,

$OA=OC$　　$2x_3x_5+2x_2x_4-x_3^2-x_2^2=0$,

$DO=AO$　　$x_7^2-2x_5x_7+x_6^2-2x_4x_6=0$,

G、A、B 三点共线 $x_1x_9 = 0$,
$GD \perp AB$ $-x_1x_8 + x_1x_6 = 0$,
F、A、C 三点共线 $x_2x_{11} - x_3x_{10} = 0$,
$FD \perp AC$ $-x_3x_{11} - x_2x_{10} + x_3x_7 + x_2x_6 = 0$,
E、B、C 三点共线 $(x_2 - x_1)x_{13} - x_3x_{12} + x_1x_3 = 0$,
$ED \perp BC$ $-x_3x_{13} + (-x_2 + x_1)x_{12} + x_3x_7 + (x_2 - x_1)x_6 = 0$.
命题结论:
G、F、E 三点共线: $(x_{10} - x_8)x_{13} + (-x_{11} + x_9)x_{12} + x_8x_{11} - x_9x_{10} = 0$.
三角化后的假设:
$\mathrm{AS} = \{(x_2 - x_1)x_{13} - x_3x_{12} + x_1x_3, (x_3^2 + x_2^2 - 2x_1x_2 + x_1^2)x_{12} + (-x_2 + x_1)x_3x_7 + (-x_2^2 + 2x_1x_2 - x_1^2)x_6 - x_1x_3^2, x_2x_{11} - x_3x_{10}, (x_3^2 + x_2^2)x_{10} - x_2x_3x_7 - x_2^2x_6, x_9, x_8 - x_6, x_7^2 - 2x_5x_7 + x_6^2 - 2x_4x_6, 2x_3x_5 + 2x_2x_4 - x_3^2 - x_2^2, 2x_4 - x_1\}$.
最后不难证明 $prem(C, \mathrm{AS}) = 0$. 也就是说 Simson 定理在适当的非退化条件下正确.

5.1.2 形式 Ⅱ

设 $\mathrm{HS} = \{H_1 = 0, \cdots, H_r = 0\}$ 是一个几何命题的假设所对应的多项式方程组,$C(x_1, x_2, \cdots, x_n) = 0$ 为几何命题的结论,$\mathrm{DS} = \{d_1 \neq 0, \cdots, d_s \neq 0\}$ 是几何命题的一组非退化条件. 令 $\mathrm{PS} = \{H_1, H_2, \cdots, H_r\}$,则由吴-Ritt 零点分解定理

$$\mathrm{Zero}(\mathrm{PS}/D) = \bigcup_{i=1}^{t} \mathrm{Zero}(\mathrm{AS}_i/DJ_i)$$

其中 D 是多项式 d_i 的乘积;AS_i 是不可约升列;J_i 是 AS_i 中多项式的初式的乘积. 如果 $t > 1$,则原几何命题为可约的. 直观地讲,也就是说几何命题的图形由若干部分具有不同性质的形式组成.

定理 1 使用上面的符号,有

(1) 命题的结论 $C = 0$ 在零点集 $\mathrm{Zero}(\mathrm{AS}_i/DJ_i)$ 所对应的图形上正确,当且仅当 $\mathrm{prem}(C, \mathrm{AS}_i) = 0$.

(2) 命题的结论 $C = 0$ 在条件 HS 与 DS 下正确,当且仅当 $\mathrm{prem}(C, \mathrm{AS}_i) = 0$, $i = 1, 2, \cdots, m$.

例 2 设 $ABDE$ 和 $CAFG$ 是分别以三角形 ABC 的边 AB 与 CA 为一边所作的两个正方形. 证明 EC 垂直于 BF.

证明 取坐标 $B:(0,0)$, $C:(x_1,0)$, $A:(x_2,x_3)$, $E:(x_4,x_5)$, $F:(x_6,x_7)$,则命题的假设与结论可写为如下代数方程.

$AB = AE$: $h_1 = x_5^2 - 2x_3x_5 + x_4^2 - 2x_2x_4$,
$AB \perp AE$: $h_2 = x_3x_5 + x_2x_4 - x_3^2 - x_2^2$,
$AC = AF$: $h_3 = x_7^2 - 2x_3x_7 + x_6^2 - 2x_2x_6 + 2x_1x_2 - x_1^2$,
$AC \perp AF$: $h_4 = x_3x_7 + (x_2 - x_1)x_6 - x_3^2 - x_2^2 + x_1x_2$.
命题结论:

$EC \perp BF$:　　$c = x_7x_5 + (x_4 - x_1)x_6 = 0.$

非退化条件：

$$A \neq B:\quad d_1 = x_2^2 + x_3^2 \neq 0,$$
$$A \neq C:\quad d_2 = (x_2 - x_1)^2 + x_3^2 \neq 0,$$
$$B \neq C:\quad d_3 = x_1 \neq 0.$$

利用吴分解原理，通过计算可得分解：

$$\mathrm{Zero}(HS/DS) = (\mathrm{Zero}(AS_1/I_1) \cup \mathrm{Zero}(AS_2/I_2) \cup \mathrm{Zero}(AS_3/I_3) \cup \mathrm{Zero}(AS_4/I_4),$$

其中

$$AS_1 = \{x_4 + x_3 - x_2, x_3x_5 - x_3^2 - x_2x_3, x_6 - x_3 - x_2, x_3x_7 - x_3^2 + (x_2 - x_1)x_3\},$$
$$AS_2 = \{x_4 - x_3 - x_2, x_3x_5 + x_2x_4 - x_3^2 - x_2^2, x_6 + x_3 - x_2, x_3x_7 - x_3^2 + (-x_2 + x_1)x_3\},$$
$$AS_3 = \{x_4 + x_3 - x_2, x_3x_5 - x_3^2 - x_2x_3, x_6 + x_3 - x_2, x_3x_7 - x_3^2 + (-x_2 + x_1)x_3\},$$
$$AS_4 = \{x_4 - x_3 - x_2, x_3x_5 + x_2x_4 - x_3^2 - x_2^2, x_6 - x_3 - x_2, x_3x_7 + (x_2 - x_1)x_6 - x_3^2 - x_2^2 + x_1x_2\}.$$

不难看出这四个分支分别对应于下面四种图形：通过计算，$\mathrm{prem}(c, AS_1) = \mathrm{prem}(c, AS_2) = 0$；$\mathrm{prem}(c, AS_3) = \mathrm{prem}(c, AS_4) \neq 0$. 也就是说在条件 $A \neq B, A \neq C, B \neq C$ 下，上面几何命题在前两个图形上正确；而在后两个图形上一般不正确. 实际上证明了"设 $ABDE$ 和 $CAFG$ 是分别以三角形 ABC 的边 AB 与 CA 为一边同时向内或向外所作的两个正方形. 则 EC 垂直于 BF".

5.2　几何公式的自动推导

吴-Ritt 分解定理不仅可以证明定理，还可以自动发现定理. 原理如下：设 AS 为分解中得到的升列，而 $A_1(u_1, u_2, \cdots, u_q, y_1)$ 为 AS 中第一个多项式，则实际上是求得了独立变量 U 与变量 y_1 之间的关系. 在几何中这种问题经常遇到，例如，几何公式的推导与轨迹方程的计算就是这样两类问题. 本节将简要介绍怎样用吴方法来处理这种问题. 详见文献[47,48].

例 2(Heron-秦公式)　试找出三角形 ABC 的面积与其三边边长之间的关系.

假设三角形三边的长度分别为 a、b、c，面积为 s，$A = (0,0)$，$B = (x_1, 0)$，$C = (x_2, x_3)$. 则有如下的几何关系：

$$h_1 = 2s - x_1x_3 = 0 \qquad (s = \text{面积})$$
$$h_1 = x_1 - c = 0 \qquad (AB = c)$$
$$h_3 = x_2^2 + x_3^2 - b^2 = 0 \qquad (AC = b)$$
$$h_4 = (x_1 - x_2)^2 + x_3^2 - a^2 = 0 \quad (BC = a)$$

令 $PS = \{h_1, h_2, h_3, h_4\}$，并在序关系 $s < a < b < c < x_1 < x_2 < x_3$ 下对 PS 进行整序. 可得 PS 的特征列 CS：

$$c_1 = 2a^2b^2 - c^4 - 16s^2 + 2b^2c^2 - b^4 + 2a^2c^2 - a^4,$$
$$c_2 = x_1 - c,$$

$$c_3 = c^2 + b^2 - a^2 - 2c \cdot x_2,$$
$$c_4 = 2 \cdot s - c \cdot x_3.$$

由于 c_1 中只含有 a、b、c 和 s，所以 $c_1 = 0$ 给出了三角形三边边长与面积之间的关系.

例3(Peaucellier 连杆问题) 如图5-1所示，设连杆 AB、AD、DC、BC 有相同的长度，EA 和 EC 有相同的长度，并假设连杆 FD 的长度与 E 到 F 的距离相同. 如果 E、F 是两固定的点，并且连杆可以绕其转动，试问结点 B 有什么样的轨迹.

设 $F = (0,0)$, $E = (r,0)$, $C = (x_2, y_2)$, $D = (x_1, y_1)$, $B = (x, y)$, n 和 m 分别为 AD 在 EDB 及与其相垂直方向上的投影长度，则几何关系可以描述为

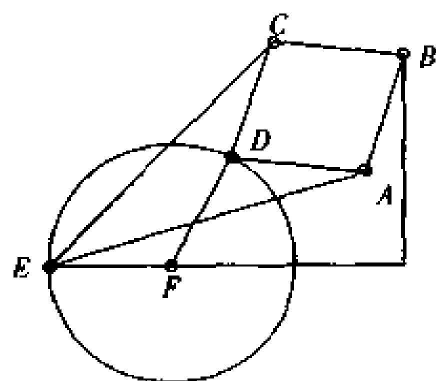


图 5-1

$$h_1 = y_1^2 + x_1^2 - r^2 = 0,$$
$$h_2 = y_2^2 - 2y_1y_2 + x_2^2 - 2x_1x_2 + y_1^2 + x_1^2 - n^2 - m^2 = 0,$$
$$h_3 = y_2^2 - 2y_1y_2 + x_2^2 - 2x_1x_2 + y_1^2 + x_1^2 - n^2 - m^2 = 0,$$
$$h_4 = y_2^2 + x_2^2 - 2x_1x_2 + y_1^2 + x_1^2 - n^2 - m^2 = 0,$$
$$h_5 = (x - r)y_1 - yx_1 + ry = 0,$$
$$d_1 = x_1 - x \neq 0.$$

选取序关系 $x < x_1 < y_1 < y_2$ 并利用本篇1章中的方法对 Zero(H/D) 作分解，可以发现 Zero(H/D) 只有一个非空分支，对应于升列：

$x + 2h + r$, $x_2 - 2rx + \cdots$, $(x - r)y_1 - yx_1 + ry$, $(4x^2 - 8rx) + \cdots$, $2yy_2 + \cdots$.

因此 B 的坐标 x、y 所满足的关系式为 $x + 2n + r = 0$. 这说明 B 的轨迹为一平行于 y 轴的直线.

其实上述想法可以推广致更一般的形式. 给定一个复数域(实数域，微分域)上的一阶谓词逻辑公式

$$f = Q_1 x_{s_1} \cdots Q_k x_{s_k} (\varphi),$$

其中 Q_j 是谓词 $\exists$ 或者 $\forall$，φ 是一个词句. 利用1.2(2.1,3.1)节中的投影定理与量词消去法消去变量 $x_{s_i}, \cdots, x_{s_k}$，得到一个只含自由变量的公式 g，使得 f 与 g 在复数域上等价. 这实际上是发现了一个新定理 g. 限于篇幅这里不再详细介绍，有兴趣的读者请见微分几何[11]与不等式[29]中的定理自动发现.

5.3 面积方法与可读证明自动生成

借助面积证明几何定理在几何解题方法中可以说是源远流长. 用面积证明几何定理算法是由张景中、高小山、周咸青在近年给出的，见文献[49]. 这一方法可以生成几何定理的简短可读证明. 在此通过几个例题来说明这一方法. 其详细描述请

见上述文献.首先引入几个基本命题.

命题1 设A、B、C是一直线上三点,P为这一直线外一点,则$\frac{AB}{AC}=\frac{S_{PAB}}{S_{PAC}}$.其中$S_{PAB}$、$S_{PAC}$是$\triangle PAB$与$\triangle PAC$的带符号的面积.

命题2(共边定理) 设M是两条非平行直线AB和$PQ(Q\neq M)$的交点,则

$$\frac{PM}{QM}=\frac{S_{PAB}}{S_{QAB}},\quad \frac{PM}{PQ}=\frac{S_{PAB}}{S_{PAQB}},\quad \frac{QM}{PQ}=\frac{S_{QAB}}{S_{PAQB}}.$$

其中S_{PAQB}是四边形$PAQB$的面积.

命题3 如果$PQ\,/\!/\,AB$则$S_{PAB}=S_{QAB}$.

用面积法证明定理的基本步骤是使用上述几何不变量(例如面积、比例等)的基本命题,从几何命题的结论中消点.当结论中的所有点被消去后,命题的结论成为一个关于某些独立变量的表达式,于是结论成立与否就不难判断了.

例4 $ABCD$是一平行四边形,证明其两对角线AC、BD互相平分.

这一几何命题的构造型描述如下:

取自由点A、B、C;求点D;求点O.为此需要证明$\frac{AO}{CO}=1$.

第一步,消去最后作出的点O.由命题2,有

$$\frac{AO}{CO}=\frac{S_{DBA}}{S_{DCB}}.$$

第二步,消去点D.由命题3,有

$$S_{DCB}=S_{ACB}=S_{CBA},\quad S_{DBA}=S_{CBA}.$$

因此,得到以下的证明,

$$\frac{AO}{CO}=\frac{S_{DBA}}{S_{DCB}}=\frac{S_{CBA}}{S_{CBA}}=1.$$

例5(射影几何基本定理) A、B、C、D为任意四点.$E=AB\cap CD$(即E为AB与CD的交点),$F=AD\cap BC$,$G=BD\cap EF$,$H=AC\cap EF$.证明E、F与G、H形成一对共轭点对,即$GE/GF=HE/HF$.

与例4的第二步证明相似,有

$$\frac{EG}{FG}=\frac{S_{EBD}}{S_{FBD}}=\frac{S_{EBD}/S_{DEF}}{S_{FBD}/S_{DEF}}=\frac{BC/CF}{BA/AE}.$$

由命题1,命题2

$$\frac{EG}{FG}=\frac{BC/CF}{BA/AE}=\frac{S_{ABC}/S_{ACF}}{S_{ABC}/S_{ACE}}=\frac{S_{ACE}}{S_{ACF}}=\frac{HE}{HF}.$$

现在总结一下用面积证明几何定理的基本步骤.

(1) 将定理的结论用面积关系或比例表示出来.

(2) 在结论表达式中消去定理中作出的点,化简这一表达式.最后得到结论.

面积方法被推广至立体几何([49]),明可夫斯基(Minkowski)几何([49]),罗巴切夫斯基(Laboshevski)几何,黎曼(Riemann)几何([50]).基于相似观点,([51])中还提出了全角法与复数方法.在([52])中面积法与全角法与搜索方法结合用来生

成一题多解与最短证明.

5.4 其它几何定理证明方法

自吴方法发明20余年来,国内外出现了一场关于几何定理机器证明的热潮.其中微分几何与不等式的自动证明已经在2.4节与3.3节介绍过.这里简要介绍其它主要结果.

20世纪80年代中期Kapur,Kutler,Stifter,Chou等人提出了基于Groebner的方法([53]、[54]、[44]).这一方法与吴方法的适用范围相同,但一般情形下速度较吴方法慢.而且较难推广之微分几何的定理证明.

洪加威在[50]中基于吴方法提出了几何定理机器证明的单点例证法,即通过计算一个数值例题来证明几何定理.这一算法复杂度很高,未能用来证明大量几何定理.张景中、杨路等人提出了多点例证法,提高了例证法的效率[56].

Kalkbrener,张景中,杨路等人提出了基于结式的证明方法[57、58].这一方法的特点是不用因式分解即可得到完整的定理证明算法.

关于非欧几何的代数化与定理证明方法,吴文俊在[1]中作了详细论述.Chou,高小山等人用吴方法给出了超越函数的证明算法,并给出了欧几里得、明可夫斯基、罗巴切夫斯基、黎曼等九种几何的分类[59].林东岱、刘卓军将吴方法推广至有限几何[60].

White,Havel,Brudelin等人曾尝试用向量法与Braket代数证明几何定理.但未给出明确算法.用向量法证明几何定理的三种算法首先在1994年由高小山,张景中,周咸青[61],Geuber,Stifter给出.李洪波在[22]中提出了基于Clifford代数与吴方法的向量算法.这一算法不仅可以用来证明初等几何定理,还可以证明微分几何中的定理.王东明,张树功等人对这一方法做了进一步改进[62].

5.5 智能几何软件《几何专家》

几何定理自动证明早期最有影响的是Chou关于吴方法的软件[44].当然还有很多其它软件.限于篇幅,这里介绍一个功能较为齐全的软件:《几何专家》.《几何专家》是一个几何智能作图与定理自动证明软件[63].它由两部分组成,一部分是几何作图器,另一部分是定理证明器.

- 作为一个图形软件,《几何专家》可以用来精确地产生、编辑、打印各种几何图形.《几何专家》还提供了动态图形变换与板块操作.动态图形变换不仅使作图过程变得生动活泼,而且为《几何专家》的一些高级功能如动画、轨迹生成、动态数值验证提供了基础.板块操作是指对图形的一部分进行操作.这一功能对于几何变换、全等三角形和相似三角形等几何知识的理解可以起到帮助作用.

- 《几何专家》实现了这二十年来产生的大部分最具代表性的几何定理证明方法:吴方法、面积法,演绎数据库法[52]、全角法、向量与复数法与Grobner基法.

这些方法不仅可以自动证明定理,还可以产生简单、漂亮的证明.更有意义的是,《几何专家》不仅可以自动证明定理,还可以自动发现几何图形的丰富的性质.

下面举例说明《几何专家》的一些应用.

例6 (垂心定理) 证明三角形的三条高相交于一点.

下面是《几何专家》用演绎数据库方法产生的证明.

(1) $AB \perp CF$,

因为 $AD \perp BC$, (2) $\angle FDB = \angle FGB$.

(2) $\angle FDB = \angle FGB$,

因为(3)$\angle DFG$,$\angle GBD$互补(F,D,B,G共圆).

(3) $\angle DFG$、$\angle GBD$互补,

因为(4)$\angle DFG$,$\angle CED$互补,(5)$\angle GBD = \angle CED$.

(4) $\angle DFG$,$\angle CED$互补,因为(6)共圆$[C,D,E,F]$.

(5) $\angle GBD = \angle CED$,因为(7)共圆$[A,D,B,E]$.

(6) 共圆$[C,D,E,F]$,因为$EF \perp EC$,$FD \perp DC$.

(7) 共圆$[A,D,B,E]$,因为$DB \perp DA$,$EB \perp EA$.

演绎数据库方法的特点是:它不仅给出了这一定理的证明,还生成了一个几何性质的数据库.该数据库包含了所给几何图形中能由《几何专家》内部所用几何公理推导出来的所有几何性质.对于垂心定理,数据库中包含下面几何信息:共线6,垂直3,共圆6,等角24,相似三角形7,等比105.

由演绎数据库方法获得的数据库中的任何一个几何性质都可以看作"新"结果.所作的实验表明:《几何专家》不但能证明许多熟知的结果,同时能自动发现新定理.以与垂心定理相应的几何构形为例,其不动点不仅包含了垂心定理的结论,而且含有另一个常见的几何性质:$\angle EGC = \angle CGD$.数据库中还含有以下相似三角形:

$\triangle DAC \backsim \triangle DBF \backsim \triangle EAF \backsim \triangle EBC$; $\triangle DBA \backsim \triangle DFC \backsim \triangle GFA \backsim \triangle GBC$;

$\triangle EAB \backsim \triangle EFC \backsim \triangle GFB \backsim \triangle GAC$; $\triangle EDF \backsim \triangle ABF \backsim \triangle ADG \backsim \triangle EBG$;

$\triangle CAB \backsim \triangle CDE \backsim \triangle GDB \backsim \triangle GAE$; $\triangle BDE \backsim \triangle BFC \backsim \triangle GDC \backsim \triangle GFE$;

$\triangle ACF \backsim \triangle ADE \backsim \triangle GDF \backsim \triangle GCE$.

如果考虑相似三角形对的话,上面公式实际上包含42对相似三角形.

例7(Mique圆定理) $P_1P_2P_3P_4P_5$是一个五边形.$Q_i = P_{i-1}P_{i+1} \cap P_iP_{i+2}$,$M_i$ = 外接圆$Q_iP_iQ_{i-1} \cap$外接圆$Q_iP_{i+1}Q_{i+1}$(这里下标应理解为模6).证明M_1、M_2、M_3、M_4、M_5在同一圆上.称该圆为五边形的Miquel圆.

《几何专家》证明这一命题花费了3.3秒.其数据库包含672项数据.数据库中有11个圆.其中六个圆为《几何专家》推出的新结果:圆$[P_2P_5Q_3M_2M_4]$;圆$[P_1P_4Q_2M_1M_3]$;圆$[P_2P_4Q_5M_1M_4]$;圆$[P_1P_3Q_4M_3M_5]$;圆$[P_3P_5Q_1M_2M_5]$,圆$[M_1M_2M_3M_4M_5]$.最后一个圆是要证明的结论.进一步搜索数据库不难发现下面新结果:

定理2 下面十组直线每组中的三线共点,且这十个点都在Miquel圆上

$\{P_iQ_i, Q_{i-2}M_{i-2}, Q_{i+3}M_{i+3}\}$, $\{P_{i-1}M_{i-2}, P_iM_{i+1}, Q_{i-1}M_{i+2}\}$, $i = 1,2,3,4,5$.

参考文献

1 Wu W T. Basic principles of mechanical theorem proving in geometries (Part on elementary geometries). Beijing: Science Press, 1984. English translation by D. M. Wang et al. Berlin: Springer, 1994.

2 Gao X S, Chou S C. On the dimension for arbitrary ascending chains. Chinese Bull of Scis, 1993(38): 396 ~ 399

3 Chou S C, Gao X S. Ritt-Wu's decomposition algorithm and geometry theorem proving, CADE'10. In: M. E. Stickel ed. LNCS, No. 449. New York: Springer-Verlag, 1990. 207 ~ 220

4 Wu W T. On the decision problem and the mechanization of theorem-proving in elementary geometry. Scientia Sinica, 1978(21): 159 ~ 172

5 Buchberger B, Collins G E, Kutzler B. Algebraic methods for geometric reasoning. Annual Review of Computer Science, vol. 3.

6 Wu W T. On a projection theorem of quasi-varieties in elimination theory, MM-Res. Preprints, MMRC, 1989 No. 4, 40 ~ 48. Also in Chin. Ann. of Math., 1990, 11B, 220 ~ 226

7 Abhyankar S S and Bajaj C. Automatic parameterization of rational curves and surfaces Ⅲ: algebraic plane curves, comp. Aided Geo Design, 1988(5): 309 ~ 321

8 Gao X S, Chou S C. On the parameterization of algebraic curves. Applicable Algebra in Elementary Communication and Computing, 1992(3): 27 ~ 38

9 Sun X D, Wang S K, Wu K. Classification of sux-vertex-type solution of the colored Yang-Baxter Equation. J of Math Phys. 1995(36): 6043

10 Gao X S, Chou S C. Implicitization of rational parametric equations. Journal of Symbolic Computation, 1992(14): 459 ~ 470

11 Gao X S, Chou S C. On the normal parameterization of curves and surfaces. The International Journal of Computational Geometry & Applications, 1991, (1): 125 ~ 136

12 Wu W T. On Chern numbers of algebraic arieties with arbitrary singularities. Acta Math Sinica, New Series, 1987, 227 ~ 236.

13 吴文俊. 代数簇的陈示性类. 数学进展, 1965(8): 402 ~ 409

14 Semple J G, Roth L. { \ it Introduction to Algebraic} Geometry, Oxford, 1949.

15 Shi H. Chern Classes of Surface with Singularities, Proc. of Inter. Seminar on Singularity and Complex Geometry, International Press, 1994.

16 Shi H. On the Chern characters of algebraic hypersurface with arbitrary singularities, Acta Math. Sinica, new series, 1988, 4, 289—300.

17 Wu W T. On the foundation of algebraic differential geometry, M-Res. Preprints, MMRC, 1989, No. 3, 1—26.

18 Ritt J F. {it Differential Algebra}, Amer. Math. Sco. Colloquium, (1950).

19 Wu W T. A mechanization method of geometry and its applications, 4. Some theorems in planar kinematics, J. Sys. Scis & Math. Sci., 1989, 2, 97—109.

20 Chou S C and Gao X S. Automated Reasoning in Differential Geometry and Mechanics: Part I. An Improved Version of Ritt-Wu's Decomposition Algorithm, Journal of Automated Reasoning, 10: 161—172, 1993; Ⅱ. Mechanical Theorem Proving, Journal of Automated Reasoning, 10: 173—189, 1993, Kluwer Academic Publishers; Ⅲ, Mechanical Formula Derivation, IFIP Transaction on Automated Reasoning, p. 1-12, North-Holland, 1993; Ⅳ, Bertrand Curves, J. of Sys. and Math., vol. 6, p. 186—192, 1993.

21 Li Z M. Mechanical Theorem Proving of the Local Theory of the Surfaces, p. 102—120, MM-Preprints, No6, 1991.

22 Li H B. Automated Reasoning with Differential Forms, Proc. of ASCM'92, 1992.

23 Wu W T. Mechanical derivation of Newton's gravitational aws from Kepler's laws, MM-Res. Preprints, MMRC, No. 1(1987)53—61.

24 Collins G E. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, LNCS 33, Springer-Verlag, 1975, 134—183.

25 Gao X S. The Discriminant Systems of Univariate Polynomials and Their Computations, Chinese Quarterly Journal of Mathematics, 1991, No2., 1—11.

26 Yang L, Hou X R and Zeng Z B. Complete Discriminant Systems, Science in China, E series, 1996, No. 5.

27 Chou S C, Gao X S and ArnonD S. On the Mechanical Proof of Geometry Theorems Involving Inequalities, Advances in Computing Research, vol. 6, (ed. C. M. Hoffmann), o. 139—181, 1992, JAI Press INc, Greenwich, USA.

28 Mcphee N, Chou S C and Gao X S. A Combination of Ritt-Wu's Method and Collins' Method, Proc. of CADE-12, France, 1994, 401—415.

29 Yang L. Practical automated reasoning on inequalities: Generic programs for inequality proving and discovering, Proceedings of the Third Asian Technology Conference in Mathmatics, W.-C. Yang et al. (eds), Springer-Verlag 1998, 24—35.

30 Wu W T. A mechanization method of geometry and its applications, 3. Mechani-cal proving of polynomial inequalities and equations-solving, MM-Res. Preprints, MMRC, (1987), No. 2, 1—17.

31 Gao X S and Chou S C, A Zero Structure Theorem for Differential Parametric Systems, Journal of Symbolic Computation, 1994, vol. 16, 585—595.

32 Yang L. On the P3P problem, MM-Preprints, 1998, No. 17.

33 Yang L, Zhang J Z and Hou X R, Non-linear Algebraic Equations and Automated Theorem Proving, ShangHai Science and Education Pub., 1996.

34 Wu W T. On surface-fitting problem in CAGD. MM-Res. Preprints, MMRC, No. 10 (1993)1—10.

35 Wu W T. Central configurauons in planet motions and vortex otions, MM-Res. Preprints, MMRC, 1995, No. 13, 1—14.

36 Shi H and Zou F M. The Flat Central Configuartions of Four Planet Motions, MM-Ppreprints, Nil5, 1997, 99—112.

37 Wu W T. A mechanization method of geometry and its applications, 6. Solving inverse kinematic equations of Puma-type robots, MM-Res. Preprints, MMRC, 1989, 4, 49—53.

38 Wu W D and Huang Y Z. Theh Direct Kinematic Solution of The Planar Steward Platform with Coplanar Ground Points, J. of Computational Mathematics, 1996, Vol. 14, Bo. 3, 263—272.

39 Liang C G. Displacement Analysis of the 6-6 Steward Platform Mechanism, Mech. Mach. Theory, 1994, vol. 29, 547—558.

41 Sendra JR and Winkler F. Symbolic Parameterization Curves, J. or Symbolic Computation, 1991, 12(6), 607—632.

Shi H. Solving of the Yang-Baxter Equation, Proc. of IWMM92, International A cademic Pub., 1992.

42 Li Z and Shi H. Exact Solution for Belousov-Zhabotinskii Reation-Diffusion System, MM-Preprints, 1994, No. 11, 39—44.

43 Li Z and Wang M. Travelling wave solutions to the two-dimensional KdV-Burgers equation. Journal of Physics A: Mathematical and General 1993, 26, 6027—6029.

44 Chou S C. Mechanical Geometry Theorem Proving, D. Reidel Pub. Company, Dordrecht, 1988.

45 Chou S C and Gao X S. Proving Constructive Geometry Statements, Cade11, D. Kapur (eds), pp. 20—34, Lect. Notes on Comp. Sci., No. 607, Springer-Verlag, 1992.

46 Wang D M and Hu S. Mechanical Theorem Proving for Constructive Geometry Theorems, J. Sys. and Math., 1987, 7.

47 Wu W T. A mechanization method of geometry and its applications, I. Distances, areas, and volumes, Sys. Sci. & Math. Scis., 1986, 6, 204—216.

48 Chou S C. and Gao X S. Mechanical Formula Derivation in Elementary Geometries, Proc. ISSAC-90, ACM, New York, 1990, pp. 265—270.

49 Chou S C, Gao X S and Zhang J Z. Machine Proof in Geometry. Singapore: World Scientific Publishing Co., 1994.

50 Yang L, Gao X S, Chou S C and Zhang Z J. Automated Proving and Discovering of Theorems in Non-Euclidean Geometries, in: Automated Deduction in Geometry, (D. \ Wang, ed.), LNAI 1360, Springer-Verlag, Berlin Heidelberg, 1998, 171—188.

51 Chou S C, Gao X S and Zhang J Z. Automated Generation of of Readable Proofs with Geometric Invariants, I. Multiple and Shortest Proof Generation, J. of Automated Reasoning, 1996, 17, 325—347.

52 Chou S C, Gao X S and Zhang J Z. A Geometry Deductive Database, (accepted by J. of Automated Reasoning).

54 Kutzler B and Stifter S. Automated Geometry Theorem Proving Using Buchberger's Algorithm, Proc. of SYMSAC'86, Waterloo, 1986, 209—214.

55 Hong J W. Can a Geometry Theorem Be Proved by an Example?, Scientia Sinica, 29, 824—834, 1986.

56 Zhang J Z, Yang L and Deng M. The Parallel Numerical Method of Mechanical Theorem Proving, Theoretical Computer Science, 1990, 74, 253—271.

57 D. Kapur, Geometry Theorem Proving Using Hilbert's Nullstellensatz, in Proc. of SYMSAC86, 202—208, ACM Press, 1986.

58 L, Yang J Z Zhang and X R. Hou A Criterion of Dependency Between Algebraic Equations and Its Applications, Proc. of the 1992 international Workshop on Mechanization of Mathematics, p. 110—134, Inter. Academic Publishers.

59 Gao X S. Transcendental Functions and Mechanical Theorem Proving in Elementary Geometries, Journal of Automated Reasoning, 6: 403—417, 1990, Kluwer Academic Publishers.

60 Lin D D and Liu Z J. Some Results in Theorem Proving in Finite Geometry, Proc. of IWMM, Inter. Academic Pub., 1992.

61 Chou S C, Gao X S and Zhang J Z, Mechanical Geometry Theorem Proving by Vector Calculation, Proc. of ISSAC-93, ACM Press, Kiev, 284—291.

62 Wang D Mechanical manipulation for a class of differential systems, J. Symbolic Computation, 1991, 12, 233—254.

63 X S, Gao Zhang J Z and S. C. Chou, Geometry Expert, Nine Chapters Pub, to appear, (in Chinese).

·计算机数学卷·

第 17 篇

符号计算

编　者　吴文达　刘卓军
审校者　高小山

目　录

引　言

符号计算又称计算机代数，是目前正在蓬勃发展的年轻领域．它具有数学和计算机科学交叉的明显特征．和数值计算的本质差别是符号计算关注准确的计算和公式推导．这一点与人类长久以来在生活和生产实践中依靠并大量应用逻辑思维和数学推导的习惯是一致的．因而符号计算是特别需要构造性和算法化的数学．其结果是促成和发展了计算机以精确的方式处理数学表达式的能力．在下面会看到，这种能力对应用领域乃至数学科学本身提供了日臻完善的工具．

从数学的角度来看，符号计算关注的一些问题也为其它的数学分支（如代数学）所关注．然而，这里尤其注重解答的算法化．譬如，多项式理想的有限生成，早已为希尔伯特(Hilbert)有限基定理所解决．但是这样的结果并不好用．特别地，理想成员问题无法据此解答．Gröbner 基的理论的出现告诉人们可以从给定的多项式理想出发，构造出特殊的有限生成基，即 Gröbner 基，简记为 GB．利用 GB 可以圆满地解决理想成员的判定问题．本篇将较为详细地介绍 GB 的理论、算法及其应用．GB 从理想的观点看待一组多项式集合．而吴方法则从代数几何的观点看待多项式组．吴方法也是近年来发展起来的精美的理论，有着许多应用．因为本卷辟有专门一篇介绍，此处不再重复．符号计算中的算法研究并不满足于能行性，还要进一步考虑它的有效性．正因为如此，一些似乎很初等很简单的问题，在这里重又得到广泛而深刻的讨论．其中求多项式的最大公因子、分解因式是典型的问题．符号计算还关注各类结式的计算、根的分离、判别式的研究等．

论及符号计算时，不能不谈到符号计算软件，近几十年发展起来的众多的符号计算软件不但为符号计算提供了可用的工具，而且在这类软件的研制过程中，还不断地提出有待符号计算专家回答的具有挑战性的数学问题．

本篇将通俗有序地介绍如上题目．

1　符号计算的特征与软件工具

在一定意义上，符号计算是相对于数值计算而存在和发展的．符号计算的界定有许多不同的观点，例如：

(1) 计算机以符号而非数值的方式加工处理数学表达式([1])．

(2) 符号计算关注代数算法的设计、分析、实现和应用([2])．

(3) 符号计算主要用来处理冗长的数学计算和推导([3])．

克努斯(D. E. Knuth) 的经典著作“计算机程序设计方法”第 2 卷([4])，更是以半数值算法这样的副标题来刻画符号计算．

1.1　符号计算的特征

人类最早适应的是精确计算，只是当计算过于复杂而人力不可及时，近似计算才被接受.计算机的出现无疑提高了数学工作者和全人类的计算能力.然而，由于计算机字长限制这样一些明显的原因，加之缺乏相应软件支持，人们作为工具对计算机的依赖在初期仍然体现于数值计算上.例如 64 位字长的计算机，能够处理的十进制数最大者已超过 10^{19}.这样的计算精度已能满足许多应用需求.但是数值计算的缺憾一直困惑着人们.1961 年，伟津森(Wilkinson) 分析了一个著名的例子：

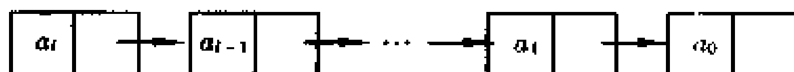
$$\prod_{i=1}^{20}(x+i)=0$$

是一个左端为 20 次的多项式方程.显然其根均是整数.若对左端 x 的 19 次方的系数做很小的扰动，比如加上 2^{-23}，其结果就完全变化了，经扰动后的方程竟然包含复数为其根.显然这个扰动后的方程能够在字长为 64 的计算机上表述出来.另一方面，不断增加计算机的字长显然是更广泛的应用所期待的.几年前，国际上一组研究人员在检验信息系统的加密能力时，用了半年多的时间，去分解由 2 个素数的乘积构成的一个 129 位字长的十进制数而获成功.为了能够解答这个问题，首先要能够在计算机上表示这个数.这有两种途径做到此，一是选字位更长的计算机，另一个是改变计算精度受囿于计算机字长的局面.后者较之前者无论从经济和通用性角度来看都更具吸引力.符号计算主要是关注后者.即能够发展无穷精度的计算环境.强调和依赖于无穷精度计算是符号计算的重要特征.

大家知道，在处理问题时可以取不同的进位基数，如 2,16,60 等.人们习惯于十进制数，而计算机则以二进制数来表达待处理的对象.不同进制数之间的互相转化是早就熟知的事.与此同时，计算机科学的进步还教会了人们一个概念，指针和链表.综合起来，并考虑到存储空间及伴随而来的计算速度等因素，在 64 位字长的计算机上取 $B=2^{60}$ 为进位基数，于是任意的一个大整数 N，都可以写成

$$N=\sum_{i=0}^{l}a_iB^i,$$

其中 $|a_i|<B$，$|a_l|\neq 0$，并且 l 刻画了数 N 的“大小”规模.如果可以用连续的地址空间表达 N 的话，l 个计算机字是必须的.否则，如下的方式需 $2l$ 个计算机字来表达 N：



这种描述方式称为链表，可以由多种方式实现.例如许多计算机语言具有指针类型数据，它是很好用的工具.为了今后叙述的方便，也是符号计算软件实现时通常采用的技术方式.总结提炼出表的概念，例如，前面提到的大整数，可以(记为或) 描述成

$$N = (a_l, a_{l-1}, \cdots, a_0).$$

这种把同样或不同样类型的一些数据排列起来，书写上左右用圆括号括起来的方式就称为表.

1.2 精确计算和软件工具

符号计算基本上不去回答韦津森分析的那类问题.那么都关注哪类问题呢?考虑二次方程

$$ax^2 + bx + c = 0 (a \neq 0). \tag{1-1}$$

大家都知道它的根是

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{1-2}$$

如果一个软件具有由(1-1)式得到(1-2)式的功能，就显示了其具有某种符号计算的能力.这种符号计算的特征带给人们的好处是可以获得精确的计算结果.换句话，每当要解二次方程(1-1)时，先是通过符号计算得到公式(1-2)，然后将 a、b、c 的值代入，来计算 x，而不是通过近似计算的方式，直接从方程(1-1)去近似 x 的值.也许解二次方程的问题过于简单，还不足以说明符号计算的意义.下面看一个较为复杂一点的例子.考虑两个首项系数为 1，其余系数为参数的 3 次多项式:

$$x^3 + a_1x^2 + b_1x + c_1 \quad 和\ x^3 + a_2x^2 + b_2x + c_2,$$

这样两个多项式，在什么样的条件下有公因子?

对于具体给定的两个多项式，可以利用欧几里得算法做辗转相除，来检验它们是否有公因子.而另一方面，符号计算则允许对任意一对这样的方程，只要检查

$$\begin{aligned} r = {} & c_2^3 - c_1^3 + 3c_2^2a_1b_1 - 2c_2^2a_2b_1 + 2b_2a_1c_1^2 - 3b_2a_2c_1^2 - b_2a_1c_2^2 - b_2^2a_1^2c_1 + \\ & b_2^2b_1c_2 + 2b_2^2b_1c_1 - b_2b_1^2c_1 + a_2a_1^2c_2^2 - 3c_2^2c_1 + 3c_2c_1^2 - b_2^3c_1 + c_2b_1^3 + \\ & a_2^3c_1^2 - c_2^2a_1^3 - 2a_2^2a_1c_1c_2 - a_2^2a_1c_1^2 + b_2^2a_1a_2c_1 + b_2a_1^2c_2b_1 - b_2a_1a_2b_1c_2 - \\ & a_1b_2c_1c_2 + 3c_2b_2a_2c_1 - 3c_1c_2a_1b_1 + c_1c_2a_2b_1 + 2a_2a_1^2c_2c_1 + a_2a_1b_2b_1c_1 - \\ & a_2a_1c_2b_1^2 + a_2^2c_2b_1^2 - 2b_2c_2b_1^2 + b_1a_2c_1^2 - a_2^2b_2b_1c_1 \end{aligned}$$

是否为0.若 $r = 0$，则它们有公因子，否则互素.虽然 r 是个相对说来较为复杂的表达式，但对于检验是否互素的问题来说，$r = 0$ 却是一个通用的判别准则，而 r 的获取是通过结式方法实现的.对上面的问题，定义

$$r = \begin{vmatrix} 1 & a_1 & b_1 & c_1 & & \\ & 1 & a_1 & b_1 & c_1 & \\ & & 1 & a_1 & b_1 & c_1 \\ 1 & a_2 & b_2 & c_2 & & \\ & 1 & a_2 & b_2 & c_2 & \\ & & 1 & a_2 & b_2 & c_2 \end{vmatrix}.$$

用纸和笔计算这个行列式不是一件愉快的事，但用符号计算软件来计算，却是轻而易举的.

最早的符号计算软件出现于 20 世纪 60 年代.例如美国 MIT 研制的 MACSYMA 是最有代表性的早期符号计算软件.它的研制计划不但雄心勃勃,而且它所用到的技术和方法深刻地影响到现今符号计算软件的风格和实现([3]).目前有许多功能丰富的符号计算软件商品,其中 Maple、Mathematica 和 Matlab 影响较大.此外,还有许多非商品化,纯粹是用于研究目的的软件系统和软件包,如 ASIR/RISA、Elmino 等.现以 Maple 为例,说明一下符号计算软件都能处理一些什么样的问题.在如下的举例中,每两行为用户与 Maple 系统的一次"对话".第一行是用户请求 Maple 处理的命令,第二行是 Maple 经处理产生的输出.因为是举例说明,这里不可能提及 Maple 的全部功能.选取的例子,其含义基本是不言自明的.

> 33!/2^31 + 41^41;

　　133087763063271199871339924096334625598993281502212891052090225051 6

> 48395254577457437347 6/122354323571234 mod 1000003;

　　887782

> sqrt(15523/3 - 98/2);

　　$124/3 \quad 3^{1/2}$

> a := sin(pi/3) * exp(2 + ln(33));

　　$a := 1/2 \quad 3^{1/2}\exp(2 + \ln(33))$

> simplify(a);

　　$32/2 \quad 3^{1/2}\exp(2)$

> evalf(a,30);

　　211.170639624855418173457016949953

由最后一个例子可以看到符号计算软件也提供浮点数的计算,而且这里还提供了无穷精度的功能.此处取精度为小数点后面 30 位.用户可以根据需要提出任意长度,如 120 位的要求.系统处理"大"数的能力不再受囿于计算机硬件的字的长度,而只依赖于系统可用的存储空间的小大.足见符号计算的魅力.

下面继续看几个有关多项式的操作,

> a := (x + y)^12 - (x - y)^12;

　　$a := (x + y)^{12} - (x - y)^{12}$

> expand(a);

　　$24yx^{11} + 440y^3x^9 + 1584y^5x^7 + 1584y^7x^5 + 440y^9x^3 + 24y^{11}x$

> factor(5 * x^4 - 4 * x^3 - 48 * x^2 + 44 * x + 3);

　　$(x - 1)(x - 3)(5x^2 + 16x + 1)$

> factor(x^6 - x^5 + x^2 + 1) mod 13;

　　$(x^3 + 10x^2 + 8x + 11)(x^3 + 2x^2 + 11x + 6)$

> factor(x^12 - y^12);

　　$(x - y)(x^2 + xy + y^2)(x + y)(y^2 - xy + x^2)(x^2 + y^2)(x^4 - x^2y^2 + y^4)$

> alias(a = Rootof(x^4 - 2)):

```
> factor(x^12 - 2*x^8 + 4*x^4 - 8, a);
```

$$(x^4-2x^2+2)(x^4+2x^2+2)(x-a)(x+a)(x^2+a^2)$$

```
> factor(x^6 - 2*x^4 + 4*x^2 - 8, a)mod 5;
```

$$(x+4)(x+2)(x+1)(x+a^2)(x+4a^2)(x+3)$$

如上的例子,已经表明符号计算软件不但可以处理整系数多项式、多变元多项式和有限域上的多项式,而且也可以处理代数扩域上的类似问题.不仅如此,符号计算软件还具有处理求极限、级数计算和微积分处理等功能.

```
> limit(tan(x)/x, x = 0);
```

$$1$$

```
> diff(ln(sec(x)), x);
```

$$\tan(x)$$

```
> series(BesselJ(o, x)/BesselJ(1, x), x, 12);
```

$$2x^{-1}-\frac{1}{4}x-\frac{1}{96}x^3-\frac{1}{1536}x^5-\frac{1}{23040}x^7-\frac{13}{4423680}x^9+O(x^{10})$$

```
> int(((3*x^2 - 7*x + 15)*exp(x) + 3*x^2 - 14)/(x - exp(x))^2, x);
```

$$\frac{3x^2-x+14}{x-\exp(x)}$$

符号计算软件的丰富功能,在下面还会看到.

1.3　精化算法:符号计算软件的发动机

符号计算软件首先需要具有无穷精度的处理能力.这些是靠前面提到的表处理技术及其它一些相关数据结构的方法来提供的.这是属于计算机软件设计这个层面的内容.理论上讲在任何一个支持无穷精度计算的可编程的环境内,都可以开发符号计算的软件系统.关键点是,并非随便拿来一个算法实现了事.可以说精化的算法能为符号计算软件提供有力的发动机.特别是一些很基础的、经常要被调用和大量被应用的算法,尤其要引起深入的关注.

例如,计算两个整数的最大公因子GCD,欧几里得算法描述简单,实现容易,可以一用.然而为了快速有效,还必须去寻求在很多人看来是微不足道的改进.

算法　二幂公因子算法([4]):设 u 和 v 是两个输入的正整数,寻找 u 和 v 的最大公因子.

步1　[决定2的幂] 令 $k\leftarrow 0$.若 u 和 v 同时为偶数,重复如下操作:

$$k\leftarrow k+1,\ u\leftarrow u/2,\ v\leftarrow v/2.$$

步2　[初始化] 如果 u 是奇数,令 $t\leftarrow -v$,并转步4.否则令 $t\leftarrow u$.

步3　[除2] 令 $t\leftarrow t/2$.

步4　[t 为偶数?] 如果 t 是偶数,返回步3.

步5　如果 $t>0$,令 $u\leftarrow t$,否则令 $v\leftarrow -t$.

步6　令 $t\leftarrow u-v$,如果 $t\neq 0$ 返回步3,否则算法停止,输出 $u\cdot u^k$.

上述算法较之欧几里得算法的表述显得很笨拙，然而它在符号计算软件里则被广泛采纳．假定给出几个很大的数，允许使用任何计算工具，比一下谁能最快地求出它们的积．这时深究一下如何做乘法的方法就是极为必要的了．一般来说，对于通用的算法，其输入必须认定是随机的．尽管如此，在做计算分析时，假定两个输入数的长度均为 n 是不过分的．传统的乘法算法需要 $O(n^2)$ 步完成乘积．而符号计算所采用的算法往往有更高的效率．

不妨设两个数是二进制表示的数，

$$u = (u_{2n-1}\cdots u_1u_0)_2,\quad v = (v_{2n-1}\cdots v_1v_0)_2.$$

令

$$U_1 = (u_{2n-1}\cdots u_n)_2,\quad U_0 = (u_{n-1}\cdots u_0)_2,$$

$$V_1 = (v_{2n-1}\cdots v_n)_2,\quad V_0 = (v_{n-1}\cdots v_0)_2,$$

于是

$$u = 2^nU_1 + U_0,\quad v = 2^nV_1 + V_0.$$

显然，

$$uv = (2^{2n} + 2^n)U_1V_1 + 2^n(U_1 - U_0)(V_0 - V_1) + (2^n + 1)U_0V_0.$$

技术上讲，乘 2 的方幂仅仅体现在移位操作上，而耗时甚微．于是上式告诉我们两个长为 $2n$ 的数之积，可以换成三个长为 n 的数之积和几个简单的移位及加法操作．如果用 $T(\quad)$ 表示该方法的计算复杂度的话，就自然有

$$T(2n) \leqslant 3T(n) + c\cdot n,$$

其中 c 是某个常数，cn 表示移位和加法的耗时．归纳有

$$T(2^k) \leqslant c(3^k - 2^k),\quad k \geqslant 1.$$

所以

$$T(n) \leqslant T(2^{\lceil \lg n\rceil} \leqslant c(3^{\lceil \lg n\rceil} - 2^{\lceil \lg n\rceil})$$

$$< 3c3^{\lg n} = 3cn^{\lg 3}.$$

因为 $\lg 3 \approx 1.586$，故 $T(n) \leqslant 3cn^{1.59}$．

按照如上架构实现的乘法算法，文献中称之为 **Karatsaua** 算法．关于乘法算法，存在其它更快的方法，见[4]．

显然符号计算的发展不但为人们提供了能力更强的工具，同时也促成相关领域一些传统结果的精益求精．另一方面，符号计算与新的数学理论，尤其是构造性的数学理论的发展的相互依赖关系变得越来越密切．

2　多项式理想与 Gröbner 基

设 K 为系数域，其特征可以为零也可以非零．K 上以 $x_1, x_2, \cdots, x_n$ 为变元的多项式环记为 $K[x_1, x_2, \cdots, x_n]$．用 PS 记 $K[x_1, x_2, \cdots, x_n]$ 内的一组多项式，

$$\mathrm{PS}\quad\begin{cases} f_1(x_1, x_2, \cdots, x_n), \\ f_2(x_1, x_2, \cdots, x_n), \\ \cdots\cdots \\ f_l(x_1, x_2, \cdots, x_n). \end{cases}$$

以 PS 为研究对象的数学分支很多，代数几何和交换代数等都发展了很深刻的

理论.

吴文俊完善起来的特征列及其整序理论(见[5])属构造性的代数几何的范畴,由于在基础和应用两个方面都有创新之举,被学术界誉为**吴方法**.吴方法是从零点集或代数簇的角度来研究 PS 的,即关注 Zero(PS) 及其结构等基本问题.因为本卷辟有专门一篇介绍吴方法及相关工作,此处不再重复.

用 Zero(PS) 记 PS 的零点集,用 Ideal(PS) 记由 PS 生成的理想,有一个明显的等式关系,

$$\mathrm{Zero}(PS) = \mathrm{Zero}(\mathrm{Ideal}(PS)).$$

下面从理想的角度关注 PS 的重要结果——**Gröbner 基**.

2.1 理想成员的判定

作为一个多项式集 PS,可以含有限多个多项式,也可能含有无穷多个多项式.对于多项式理想来说,除零理想外,均含无穷多个多项式.

理想成员问题 设 A 是 $K[x_1,x_2,\cdots,x_n]$ 上的一个理想,$g\in K[x_1,x_2,\cdots,x_n]$.问是否 $g\in A$?

因为当 A 不是零理想时,它将包含无穷多个多项式.对于给定的多项式 $g\in K[x_1,x_2,\cdots,x_n]$,要想决定是否有 $g\in A$ 绝非易事,这意味着将 g 与 A 中元素逐一对比方能回答这个问题.这样的方案显然是行不通的.因此需要一个从无限到有限的过渡.一个重要的结果建立起了这种联系.

希尔伯特基定理([6]) 对任意 $K[x_1,x_2,\cdots,x_n]$ 上的理想 A,存在 A 的有限子集,不妨记为 PS,并且 A 中的每个元素 g 都能由 PS 中元素线性组合而成.称 PS 为 A 的**生成基**,记成 $A=\mathrm{Ideal}(PS)$.

遗憾的是希尔伯特基定理的有限特性不足以解决理想成员的判定问题.对于这个问题,历史上有过一些后来发现有错误的解答.直到 1964 年,日本学者 Hironaka 证明每个多项式理想都存在一个标准基,利用该标准基可以解决理想成员的判定问题.依旧是美中不足,Hironaka 的结论是存在性的.1965 年,奥地利学者 Buchberger 在他的博士论文里给出了如何构造出这种标准基的算法,并称其为 Gröbner 基.Gröbner 是代数学家,也是 Buchberger 的老师.

算法 Buchberger:

```
Procedure Gbasis(P)
 # P 是给定的一组多项式,计算 G,使得
 # Ideal(G) = Ideal(P),并且 G 是 Gröbner 基,G ← P; k ← length(G).
 # 用 G_i 表示编好号的 G 中的第 i 个元素,B ← {[i,j] | 1 ≤ i < j ≤ k}.
while B ≠ ∅ do{
   [i,j] ← selectpair(B,G)
   B ← B - {[i,j]}
   h ← Reduce(spoly(G_i,G_j),G)
   if h ≠ 0 then {
```

$G \leftarrow G \cup \{h\}; k \leftarrow k+1$

$B \leftarrow B \cup \{[i,k], 1 \leqslant i < k\}\}$

return(G)

end

这一算法的详尽解释可以从许多文献,如[3]、[7] 中找到.我们结合一个例子对计算 Gröbner 基的一些关键点做一些解释.

考虑 $Q[x,y,z]$ 上的一组多项式

$$P = \{x^2+yz-2, y^2+xz-3, xy+z^2-5\},$$

在考虑相对应的 Gröbner 基的计算过程中,每一个多项式所含的不同的单项式,它们之间的顺序是有意义的.单项式之间的先后次序主要地是由变元之间的顺序决定.规定了变元之间的顺序,就可以进一步决定是取字典序还是取全次序等.不妨取全次序,即看两个单项式之间的“大小”时,先计算单项式中所有变量的次数和(全次数),全次数高者为大.当两个单项式的全次数相等时,要看哪个单项式具有更大的反字典序,选中的单项式为大.在以上具体的例子中,实质上已规定了单项式之间的顺序如下:

$$1 <_D z <_D y <_D x <_D z^2 <_D yz <_D xz <_D y^2 <_D xy <_D x^2 <_D \cdots$$

根据 Buchberger 算法,很重要的操作是将每个多项式按单项式由大到小排序.排在前面者,称为多项式的首项.而多项式之间的编号可以随意.于是有

$$G_1 = x^2+yz-2,$$

$$G_2 = y^2+xz-3,$$

$$G_3 = xy+z^2-5.$$

s 多项式是计算 Gröbner 基过程中最关键的概念,它定义成

$$\text{spoly}(p,z) = \text{LCM}(M(p),\quad M(q))\left[\frac{p}{M(p)} - \frac{q}{M(q)}\right],$$

其中 $M(p)$ 是指多项式 p 的首项.例如

$$\text{spoly}(G_1,G_2) = y^2(x^2+yz-2) - x^2(y^2+xz-3)$$

$$= -x^3z+y^3z+3x^2-2y^2.$$

另一个至关重要的操作是一个多项式对另一个多项式的约化.设 p 和 q 是两个多项式,假设 $M(q)$ 是 P 中某单项式 m 的因子.不妨记成

$$m = s \cdot M(q),$$

于是称 p 对于 q 可约化,并定义 p 对 q 的一次约化为 p 中 m 重写成 $s\cdot(M(q)-q)$,记成 p_1.若 p_1 仍对于 q 可约化,则继续这个过程,直到得到 p_t 不再对于 q 可约化.最后记成

$$p \mapsto_q^* p_t.$$

现在利用 Buchberger 算法记算 Gbasis(P).

第一阶段,有 $B = \{[1,2],[1,3],[2,3]\}$.由于

$$\text{spoly}(G_1,G_2) = -x^3z+y^3z+3x^2-2y^2$$

$$\mapsto_{G_1} y^3z+xyz^2+3x^2-2y^2-2xz$$

$$\begin{aligned}&\mapsto_{G_2} xyz^2 + 3x^2 - 2y^2 - 2xz + 3yz\\ &\mapsto_{G_1} - 2y^2 - 2xz + 6\\ &\mapsto_{G_2} 0\end{aligned}$$

此时,$B = \{[1,3],[2,3]\}$. 接下去有

$$\begin{aligned}\mathrm{spoly}(G_2,G_3) &= y^2z - xz^2 + 5x - 2y\\ &\mapsto_{G_2} - 2xz^2 + 5x - 2y + 3z.\end{aligned}$$

由于不能再约化,得到 $G_4 = - 2xz^2 + 5x - 2y + 3z$,并置 $k = 4$. 进入下一阶段,有

$$B = \{[2,3],[1,4],[2,4],[3,4]\}.$$

类似地可以继续下去,得到

$$\mathrm{spoly}(G_2,G_3) \mapsto_{G_1} - 2yz^2 - 3x + 5y + 2z = G_5,$$

$$\mathrm{spoly}(G_3,G_4) \mapsto_{G_1}^{*} - 2z^4 - 2xz - 3yz + 15z^2 - 19 = G_6,$$

这时有 $B = \{[1,5],[2,5],[3,5],[4,5],[1,6],[2,6],[3,6],[4,6],[5,6]\}$.
以下会发现,所有新生成的 s 多项式都对 G 约化成 0. 即算法终止时,输出的 G 为 $\{G_1, G_2, G_3, G_4, G_5, G_6\}$.

显然,$\mathrm{Ideal}(P) = \mathrm{Ideal}(G)$.

Gröbner 基的一条重要性质是说,对于任意的多项式 $g \in \mathrm{Ideal}(P)$,当且仅当 $g \mapsto_G^* 0$. 而当 P 自身不是 Gröbner 基时,一般 $g \mapsto_P^* 0$ 不成立.

根据这一性质,理想成员的判定问题获得圆满解答.

2.2　Gröbner 基的性质和应用

大家已经知道,理想成员问题正是借助 Gröbner 基的计算来解决的. 然而,特别是在几何应用当中,人们对理想成员的问题并不是特别感兴趣,倒是对于根式成员问题重视有加.

根理想成员的判定问题　设 $f, f_1, \cdots, f_m$ 是 $K[x_1, x_2, \cdots, x_n]$ 上的多项式. 问是否 $\mathrm{Zero}(f) \supseteq \mathrm{Zero}(\{f_1, f_2, \cdots, f_m\})$.

显然,

$$\begin{aligned}&\mathrm{Zero}(f) \supseteq \mathrm{Zero}(\{f_1, f_2, \cdots, f_m\})\\ &\Leftrightarrow \mathrm{Zero}(\{f_1, f_2, \cdots, f_m, z \cdot f - 1\}) = \varnothing\\ &\Leftrightarrow 1 \in \mathrm{Ideal}(\{f_1, f_2, \cdots, f_m, z \cdot f - 1\}).\end{aligned}$$

于是根理想成员的判定问题被转换成理想成员的判定问题. 这即表明用 Gröbner 基的方法也能解决几何定理机器证明问题. 参见本卷第 18 篇《数学机械化与机械化数学》.

为了拓广应用,我们对理想 $\mathrm{Ideal}(P)$ 的 Gröbner 基 G 再附加上若干条件,并赋予新的名字.

令 G 是多项式组 P 生成的理想的 Gröbner 基,

(1) 如果对所有 g、$h \in G$,并且 $g \neq h$,都有 g 不能由 h 约化,则称 G 是简化了

的 Gröbner 基；

(2) 如果对所有 $g \in G$，g 的首项系数是 1，则称 G 是正则的 Gröbner 基.

现在能够决定是否两个理想是等价的.设要判断的两个理想分别是 I 和 J，它们的正则简化 Gröbner 基是 G_I 和 G_J，则 $I = J$ 当且仅当 $G_I = G_J$.

考虑一组方程

$$f_1(x_1, x_2, \cdots, x_n) = 0,$$
$$f_2(x_1, x_2, \cdots, x_n) = 0,$$
$$\cdots\cdots$$
$$f_m(x_1, x_2, \cdots, x_n) = 0,$$

其中，$f_1, f_2, \cdots, f_m \in K[x_1, x_2, \cdots, x_n]$.用 $\overline{K}$ 记域 K 的代数闭包.要判断如上的方程组在 $\overline{K}^n$ 上是否有解.结论是：

令 $I = \mathrm{Ideal}(\{f_1, f_2, \cdots, f_m\})$，$G$ 是 I 的 Gröbner 基，方程组 $f_1 = 0, \cdots, f_m = 0$ 在 $\overline{K}^n$ 上无解，当且仅当 $1 \in G$.进一步的深刻结果是，上面的方程组在 $\overline{K}^n$ 上有解，并且解的数目有限，当且仅当对每个 $1 \leqslant i \leqslant n$，都能找到 $g_i \in G$，使得 g_i 的首项只含变元 x_i.

Gröbner 基还有一个非常漂亮的性质，称为 Gröbner 基的消去性质.设 G 是理想 I 在变元序 $x_1 < x_2 < \cdots < x_n$ 相伴字典序下的 Gröbner 基，则

$$I \cap K[x_1, \cdots, x_i] = \mathrm{Ideal}(G \cap K[x_1, \cdots, x_n]).$$

现在集中来看一些应用上述性质处理和解答的问题.

例 1　考虑方程组 $f_1 = f_2 = f_3 = 0$，其中

$$f_1 = 4xz - 4xy^2 - 16x^2 - 1,$$
$$f_2 = 2y^2z + 4x + 1,$$
$$f_3 = 2x^2z + 2y^2 + x.$$

记这组多项式为 P，显然 $P \subseteq Q[x, y, z]$.令 $\overline{Q}$ 记代数数域，要知道最初的方程组是否在 $\overline{Q}^3$ 中有解？

规定 $x < y < z$ 及单项式之间的字典序，由 P 出发，算得 Gröbner 基 $G = \{g_1, g_2, g_3\}$，其中

$$g_1 = 65z + 64x^4 - 432x^3 + 168x^2 - 354x + 104,$$
$$g_2 = 26y^2 - 16x^4 + 108x^3 - 16x^2 + 17x,$$
$$g_3 = 32x^5 - 216x^4 + 64x^3 - 42x^2 + 32x + 5.$$

由 Gröbner 基的性质知道，$f_1 = f_2 = f_3 = 0$ 在 $\overline{Q}^3$ 上有解，并且解的数目有限.

例 2　用同样的方法考虑系数中含有参数的情形.设有方程组 $f_1 = f_2 = f_3 = f_4 = 0$，其中

$$f_1 = x_4 + b - d,$$
$$f_2 = x_4 + x_3 + x_2 + x_1 - a - c - d,$$
$$f_3 = x_3x_4 + x_1x_4 + x_2x_3 - ad - ac - cd,$$
$$f_4 = x_1x_3x_4 - acd.$$

视f_1,f_2,f_3,f_4为$Q(a,b,c,d)[x_1,x_2,x_3,x_4]$上的多项式，其集合仍记为$P$.依旧规定$x_1<x_2<x_3<x_4$及字典序.算得$P$的正则且简化）的Gröbner基$G=\{g_1,g_2,g_3,g_4\}$，其中

$$g_1=x_4+b-d,$$

$$g_2=x_3-\frac{b^2-2bd+d^2}{acd}x_1^2-\frac{abc+abd-acd-ad^2+bcd-cd^2}{acd}x_1-a-c-d,$$

$$g_3=x_2+\frac{b^2-2bd+d^2}{acd}x_1^2+\frac{abc+abd-ad^2+bcd-cd^2}{acd}x_1-b+d,$$

$$g_4=x_1^3+\frac{ac+ad+cd}{b-d}x_1^2+\frac{a^2cd+ac^2d+acd^2}{(b-d)^2}x_1+\frac{a^2c^2d^2}{(b-d)^3}.$$

显然，所关注的方程组具有有限多个解.

关于Gröbner基的更多的性质和应用，许多文献都做了详尽的讨论，例如见[2]、[3]、[7].尽管如此，Gröbner基不是符号计算的全部，还有许多其它的关键技术需要考虑.这样一些问题的关注和讨论，不但丰富了符号计算的内容，同时，有些对于提高Gröbner基算法的效率也是必需的.

3　符号计算中的一些关键技术

符号计算从理论到算法到软件到应用都有了丰富的积累和显著的进步.涉及到的领域和内容也在不断扩大.从逻辑到代数，从纯代数的处理到微分代数问题的考虑，从特征为0到特征非0，从复数情形到实数情形；从数值计算和符号计算分离到混合计算的关注，从图形用户界面到可视化数学的投入等等，所有这些都可以从符号计算的国际学术大会ISSAC（Interastional Symposiam on Symbolic and Algebraic Computation）相关的学术杂志和学术论文及专著中了解到.然而，最为基础、最为完整和系统化的工作还是集中于相关于多项式的处理问题上.

3.1　结式及迪克森结式

求解代数方程组具有的理论和应用的双重价值已被广泛接受.其中，为了求解的目的，把多变元的问题逐步通过消元的方式化成依次解单变元的代数方程，是目前人们在处理此类问题时仍在遵循的基本思路.这种思考方式历史上为笛卡尔所提倡.其实更早，中国的古代数已形成了消元的算法，吴方法（[5]）是对历史的一种继承，当然是一种消元法.前面了解到的Gröbner基由于具有的消去性质，因而也可以视为一种消元法.此处，关注一下另外的消元方法.

结式的概念在前面为了判定两个多项式有否公因子的时候已有所接触.结式不仅局限于两个多项式的运算，而且可以发展成结式系统（[8]）.

用K记某个数域，$\overline{K}$记它的代数闭包.考虑$K[u_1,u_2,\cdots,u_m][x]$上一组多项

式 $f_1, f_2, \cdots, f_r$. 视这组多项式为单变元 x 的多项式，则可以构造出另一组多项式 $D_1, D_2, \cdots, D_h \in K[u_1, u_2, \cdots, u_m]$，并具有这样的性质：

$D_1 = D_2 = \cdots = D_h = 0$ 在 $\overline{K}^m$ 上有解，当且仅当 $f_1 = f_2 = \cdots = f_r = 0$ 有解，或

全部多项式 $f_1, f_2, \cdots, f_r$ 的形式首项系数为 0. 称多项式组 $D_1, D_2, \cdots, D_h$ 为多项式组 $f_1, f_2, \cdots, f_r$ 的**结式系统**.

由 f 到 D 显然是一种消元. 自然，这种消元过程可对 $D_1, D_2, \cdots, D_h$ 再次进行. 把由 f 到 D 及 f 与 D 之间关联的结果称为范 · 德瓦尔登定理. 把下面要介绍的如何由 f 构造出 D 的过程称为**克罗内克(Kronecker) 过程**.

克罗内克过程：

(1) 对 $f_1, f_2, \cdots, f_r$，记最大的次数为 n.

(2) [由 f 构造 g] 对所有的 $1 \leqslant i \leqslant r$，用 n_i 记 f_i 关于 x 的次数. 若 $n_i = n$，则 f_i 自身记成一新的 g；若 $n_i < n$，则构造两个新的 g：$x^{n-n_i}f_i$ 和 $(x-1)^{n-n_i}f_i$.

这一步结束时，得到 $g_1, g_2, \cdots, g_s$，一般说来总有 $s \geqslant r$.

(3) 引进新的不定元 u、v，并构造

$$g_u = u_1 g_1 + \cdots + u_s g_s,$$

$$g_v = v_1 g_1 + \cdots + v_s g_s.$$

可以说明 g_u 与 g_v 有公因子，当且仅当 $g_1, g_2, \cdots, g_s$ 有公因子([8]).

(4) 对 g_u、g_v 做结式 R，R 可表示成

$$\sum_{i=1}^{h} D_i u_1^* u_2^* \cdots u_s^* v_1^* v_2^* \cdots v_s^*,$$

* 代表次数.

至此，得到结式系统 $D_1, D_2, \cdots, D_h$. 由以上分析知道：

$D_1 = 0, D_2 = 0, \cdots, D_h = 0$，当且仅当 R 作为 u 和 v 的多项式恒等于 0；当且仅当 g_u 和 g_v 有公因子或所有的首项系数等于 0. 进一步可以推知，当且仅当 $f_1 = f_2 = \cdots = f_r = 0$ 有解，或它们的所有首项系数为 0.

结式系统是个很漂亮的理论结果. 目前流行的符号计算软件均支持求结式的计算，但对于结式系统的构造尚不支持. 一定程度上讲，改进构造过程是必要的. 例如，当已知某个 f_i，不妨说 f_1 的首项系数不为 0 时，就可以用 f_1 与 $v_2 f_2 + \cdots + v_r f_r$ 的结式代替 g_u 和 g_v 的结式而有同样的结论.

迪克森(Dixon) 结式也是一种消元技术([9]). 起源于 1908 年迪克森的工作.

设给定 $k+1$ 个 k 变元多项式组

$$\text{PS} \begin{cases} p_1(x_1, x_2, \cdots, x_k), \\ p_2(x_1, x_2, \cdots, x_k), \\ \cdots\cdots \\ p_{k+1}(x_1, x_2, \cdots, x_k). \end{cases}$$

设 $\alpha_1, \alpha_2, \cdots, \alpha_k$ 是 k 个新变元，则如下行列式

$$\Delta(x_1,\cdots,x_k,\alpha_1,\cdots,\alpha_k)=\begin{vmatrix} p_1(x_1,x_2,\cdots,x_k) & \cdots & p_{k+1}(x_1,x_2,\cdots,x_k) \\ p_1(\alpha_1,x_2,\cdots,x_k) & \cdots & p_{k+1}(\alpha_1,x_2,\cdots,x_k) \\ p_1(\alpha_1,\alpha_2,\cdots,x_k) & \cdots & p_{k+1}(\alpha_1,\alpha_2,\cdots,x_k) \\ \vdots & & \vdots \\ p_1(\alpha_1,\alpha_2,\cdots,\alpha_k) & \cdots & p_{k+1}(\alpha_1,\alpha_2,\cdots,\alpha_k) \end{vmatrix}$$

是关于变元 x_i、α_i 的多项式，并且当 $x_i=\alpha_i(i=1,\cdots,k)$ 时 $\Delta(x_1,\cdots,x_k,\alpha_1,\cdots,\alpha_k)=0$. 这意味着 $(x_1-\alpha_1)\cdots(x_k-\alpha_k)$ 是 $\Delta(x_1,\cdots,x_k,\alpha_1,\cdots,\alpha_k)$ 的因子. 令

$$\delta(x_1,\cdots,x_k,\alpha_1,\cdots,\alpha_k)=\frac{\Delta(x_1,\cdots,x_k,\alpha_1,\cdots,\alpha_k)}{(x_1-\alpha_1)(x_2-\alpha_2)\cdots(x_k-\alpha_k)},$$

称之为 $p_1,p_2,\cdots,p_{k+1}$ 的迪克森多项式. 进一步可视迪克森多项式为 $\alpha_1,\alpha_2,\cdots,\alpha_k$ 的多项式，并把 $\alpha_1,\alpha_2,\cdots,\alpha_k$ 的各不同幂积的系数记为

$$c_i(x_1,x_2,\cdots,x_k),\quad i=1,2,\cdots,n\ (n\text{ 是个待定的数}),$$

称方程组

$$c_i(x_1,x_2,\cdots,x_k)=0,\quad i=1,2,\cdots,n$$

为方程组 $p_1=0,p_2=0,\cdots,p_{k+1}=0$ 的**迪克森导出方程组**. 易知原方程组的解必定是其迪克森导出方程组的解.

在上述迪克森导出方程组中，把其中出现的关于变元 $x_1,x_2,\cdots,x_k$ 的所有幂积按字典序由大到小排成

$$e_n,e_{n-1},\cdots,e_2,e_1,$$

于是迪克森导出方程组就可以写为标准形式

$$\boldsymbol{D}\begin{bmatrix} e_n \\ \vdots \\ e_1 \end{bmatrix}=\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

其中 $\boldsymbol{D}$ 为迪克森导出方程组关于幂积 $e_n,\cdots,e_2,e_1$ 的系数矩阵，称为多项式组 $p_1,p_2,\cdots,p_{k+1}$ 的**迪克森矩阵**. 而其行列式 $\det(\boldsymbol{D})$ 为**迪克森结式**.

对于 $k+1$ 个 k 变元的一般完全多项式，迪克森证明：迪克森结式为0是它们有仿射零点的一个必要条件.

关于一般多项式、全次数多项式、一般全次及一般完全多项式如何确定 n 的值，请参见文献[9].

现在考察一个例子. 设

$$p_1(s,t)=(s^2+s^2t)x-s^2t+t+s^2+1,$$
$$p_2(s,t)=(s^2+s^2t)y-s^2t-s+t,$$
$$p_3(s,t)=(s^2+s^2t)z-2s^2+2t+2,$$

按上述一般过程，由计算可得，

$$\Delta(s,t,\alpha,\beta)=\begin{vmatrix} p_1(s,t) & p_2(s,t) & p_3(s,t) \\ p_1(\alpha,t) & p_2(\alpha,t) & p_3(\alpha,t) \\ p_1(\alpha,\beta) & p_2(\alpha,\beta) & p_3(\alpha,\beta) \end{vmatrix}$$

$$= (s - \alpha)(t - \beta)(c_1\alpha^3 + c_2\alpha^2 + c_3\alpha + c_4).$$

于是 p_1、p_2、p_3 的迪克森多项式为

$$\delta(s,t,\alpha,\beta) = c_1\alpha^3 + c_2\alpha^2 + c_3\alpha + c_4$$

这里有

$$c_1 = (2x + 2z - 2)s + (6y - 4x - z - 2)t + 6y - 2x + z - 4,$$
$$c_2 = (6y - 4x - z - 2)st + (6y - 2x + z - 4)s + (2x - z - 2)t + 2x - z - 2,$$
$$c_3 = (2x - z - 2)st + (2x - z + 4)s + (2x - z - 2)t + 2x - z + 4,$$
$$c_4 = (2x - z - 2)st + (2x - z + 4)s.$$

由 $c_1 = 0, c_2 = 0, c_3 = 0, c_4 = 0$,可得

$$\boldsymbol{D}\begin{bmatrix} st \\ s \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

其中

$$\boldsymbol{D} = \begin{bmatrix} 0 & 2x + 2z - 2 & 6y - 4x - z - 2 & 6y - 2x + z - 4 \\ 6y - 4x - z - 2 & 6y - 2x + z - 4 & 2x - z - 2 & 2x - z - 2 \\ 2x - z - 2 & 2x - z + 4 & 2x - z - 2 & 2x - z + 4 \\ 2x - z - 2 & 2x - z + 4 & 0 & 0 \end{bmatrix}.$$

容易验证 $\det(\boldsymbol{D}) = 0$.

这种由 PS 出发到算出 $\det(\boldsymbol{D})$ 的过程,一下子能消去多个元,在探讨多项式解的过程中有着重要的应用.

3.2　根的分离与斯特姆序列

消元的目的,最根本的是化多变元多项式的处理为单变元的多项式处理.而对于单变元的代数方程来说,一般地并不存在求根公式.因此,为了把握解,近似计算的方法是无法回避的,至少需要根的分离技术.

对于一个实系数的代数方程 $f(x) = 0$,求它的实根分离,就是要在实数轴上决定出若干区间,使得 $f(x) = 0$ 在这些区间上只有一个(重)实根.为达此目的,需要能做到

(1) 能够估计出 $f(x) = 0$ 的不同根之间最小距离的下界.Collins([10])给出了这样的下界.

(2) 给定任何一个区间 $[a, b]$,都能算知 $f(x) = 0$ 在其上有几个实根.斯特姆(Sturm)序列是解答这个问题的基础.

对给定的 $f(x)$,可以构造相应的斯特姆序列如下:

$$f_0 = f(x),\quad f_1 = f'(x),\quad f_2, \cdots, f_l,$$

其中

$$f_i = q_{i-1}(x) \cdot f_{i-1}(x) - f_{i-2}(x),\quad i \geq 2.$$

对于任意的 $c \in \mathbf{R}, f(c) \neq 0$,令 $V(c)$ 为序列

$$\langle f_0(c), f_1(c), \cdots, f_l(c)\rangle$$

的符号变号数. 当 a、$b \in \mathbf{R}$, 且 $f(a)\cdot f(b) \neq 0$ 时, 区间 $[a,b]$ 上 $f(x)=0$ 不同根的个数等于 $V(a)-V(b)$.

至于复根的分离, 注意到

$$f(z) \to f(x+\mathrm{i}y) = f_1(x,y) + \mathrm{i}f_2(x,y),$$

可以通过对

$$R_1(x) = \mathrm{Resultant}(f_1, f_2, y)$$

和

$$R_2(y) = \mathrm{Resultant}(f_1, f_2, x)$$

同时做实根分离, 来分析出其复根的分离状况. 最近, 文献[9] 推广了斯特姆序列的方法, 得了多项式根的完全判别系统.

下面以一个有趣的问题结束本节的讨论. 如何决定复系数多项式的稳定性条件(文献[11]). 说 $f(z)$ 是稳定的, 如果它的全部根都位于左半平面上. 由 $f(z)$ 构造

$$f(z_1 + \mathrm{i}z_2) = f_1(z_1, z_2) + \mathrm{i}f_2(z_1, z_2),$$

由结式运算得

$$R(z_1) = \mathrm{Resultant}(f_1, f_2, z_2).$$

结论是 $f(z)$ 是稳定的, 当且仅当

(1) $R(z_1)$ 的常数项非 0.

(2) $R(z_1)$ 的全部非零系数符号相同.

例如 $f(z) = z^4 + 3z^2 + 7z^2 + 16z + 12$, 经计算得

$$R(z_1) = 16z_1^2(z_1^4 + 3z_1^3 + 7z_1^2 + 16z_1 + 12)\cdot (15z_1^5 + 96z_1^4 + 248z_1^3 + 352z_1^2 + 289z_1 + 130)^2.$$

由于 $R(z_1)$ 的常数项是 0, 故 $f(z)$ 不稳定.

再看一个例子

$$f(z) = z^3 + (\mathrm{i}2 + \sqrt{3} + 3\sqrt{2})z^2 + (\mathrm{i}2\sqrt{3} + \mathrm{i}\sqrt{2} + 3\sqrt{2}\sqrt{3} + 7)z + (\mathrm{i}\sqrt{2}\sqrt{3} + 7\sqrt{3}),$$

计算得

$$\begin{aligned} R(z_1) = {} & 64z_1^9 + (192\sqrt{3} + 576\sqrt{2})z_1^8 + (1632\sqrt{2}\sqrt{3} + 5504)z_1^7 + \\ & (13152\sqrt{3} + 17408\sqrt{2})z_1^6 + (32112\sqrt{2}\sqrt{3} + 77296)z_1^5 + \\ & (102720\sqrt{3} + 120976\sqrt{2})z_1^4 + (110160\sqrt{2}\sqrt{3} + 258432)z_1^3 + \\ & (151920\sqrt{3} + 180800\sqrt{2})z_1^2 + (61104\sqrt{2}\sqrt{3} + 148992)z_1 + \\ & 22848\sqrt{3} + 26112\sqrt{2}. \end{aligned}$$

根据判别条件, 易知 $f(z)$ 是稳定的多项式.

3.3　模运算与中国剩余定理

符号计算中一个异常引人注目的现象是中间计算结果的快速膨胀. 例如, 当利用计算来判断两个多项式 f 和 g 是否互素时, 这样的现象尤为严重. 因为, 当二者互

素时，辗转相除的算法应能告之它们的公因子是常数. 实际运行中，往往在得到这个常数之前，处理和加工的表达式已经变得很大了.

例如，考虑

$$f(x) = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5,$$
$$g(x) = 3x^6 + 5x^4 - 4x^2 - 9x + 21,$$

在有理数域上使用欧几里得算法，得

$$R_2(x) = -\frac{5}{9}x^4 + \frac{1}{9}x^2 - \frac{1}{3},$$
$$R_3(x) = -\frac{117}{25}x^2 - 9x + \frac{411}{25},$$
$$R_4(x) = \frac{233150}{19773}x - \frac{102500}{6591},$$
$$R_5(x) = -\frac{1288744821}{543589225}.$$

如果应用欧几里得算法到 $z[x_1, x_2, \cdots, x_n]$，情形会更糟糕.

有几种途径改进这一状况. 一是利用子结式序列的计算技巧，另一个是采用模算法. 所谓模算法是设 R 和 R' 均是唯一可分解环，选取一个环同态映射 $\varphi: R \to R'$.

假设 $f(x)$、$g(x) \in R[x]$ 并且 $C(x) = \mathrm{GCD}(f(x), g(x))$. 若有 $\varphi(\mathrm{lcoef}(C(x))) \neq 0$，则

$$\deg(\mathrm{GCD}(\varphi(f(x)), \varphi(g(x)))) \geqslant \deg(\mathrm{GCD}(f(x), g(x))).$$

典型的这种映射有两类：

(1) 单变元情形，多是整数环到整数剩余类环；

(2) 多变元情形，将某个变元映射到一个具体值上.

对于多变元情形，有时(1) 和(2) 被组合使用.

例如，仍考虑前面的 $f(x)$、$g(x)$ 和映射 $\varphi: z[x] \to z_2[x]$，有

$$\varphi(f(x)) = x^8 + x^6 + x^4 + x^3 + x^2 + 1,$$
$$\varphi(g(x)) = x^6 + x^4 + x + 1.$$

在 $z_2[x]$ 内计算得

$$\mathrm{GCD}(\varphi(f(x)), \varphi(g(x))) = x + 1.$$

我们知道，原始的多项式是互素的，而取模后算出了公因子. 这表明这个具体的模映射取得不好，称之为不幸运的同态映射. 幸运的是，这种不幸运的映射数目很少. 假如取到一个同态映射，使得同态像的公因子是常数，即意味着原始的多项式是互素的，这是由同态映射的性质保证的.

如果映射 φ 不是一个不幸运的同态映射，容易证明存在一个常数 c，满足

$$\varphi(\mathrm{GCD}(f(x), g(x))) = c \cdot \mathrm{GCD}(\varphi(f(x)), \varphi(g(x))).$$

由于不幸运的映射个数不是很多，当选取一定数目的映射后，必能取到非不幸运的映射. 假设若干的映射使得计算结果相似，即次数相同，接下去就可以用中国剩余定理来恢复出真正的公因子. 例如

$$\varphi_7:\ z[x] \to z_7[x],$$
$$\varphi_{11}: z[x] \to z_{11}[x],$$

使得

$$\mathrm{GCD}(\varphi_7(f(x)), \varphi_7(g(x))) = x^2 + 1,$$
$$\mathrm{GCD}(\varphi_{11}(f(x)), \varphi_{11}(g(x))) = x^2 + 3x + 4,$$

我们就可以猜到

因为

$$\mathrm{GCD}(f(x), g(x)) = x^2 + ax + b,$$
$$a \equiv 0 \pmod 7, \quad a \equiv 3 \pmod{11},$$
$$b \equiv 1 \pmod 7, \quad b \equiv 4 \pmod{11}.$$

由中国剩余定理,可以算知

$$a = 14, \quad b = 15,$$

通过试除检验知道

$$\mathrm{GCD}(f(x), g(x)) = x^2 + 14x + 15.$$

模运算带给我们至少两个好处:一是克服了计算的中间结果快速膨胀的问题;第二是提供了平行计算的可能性.

对于多变元情形,要想从映像的计算结果恢复出原像的结果来,除了要用到中国剩余定理外,还要用到插值的技术.可以参见[3],以便欣赏更多的实例.

3.4　多项式的因式分解

多项式的因式分解当然也要用到模运算的技巧和方法.此外,由于多项式因式分解在其它符号计算的环节中起着重要作用,对这一问题加以特别的关注是极正常之事.尤其是代数扩域上的因式分解,已引发了数学上很深刻的工作.Treger 的这一贡献可以在几乎所有讲述代数扩域上因式分解的著作中找到,如[3]、[5]等.Berlekamp 在有限域上的多项式因式分解算法,也是被反复引用的工作([3][4]),所有上述讨论都可以在相关文献中找到详细的论述.下面以一些问题来结束这里的讨论.

对于单变元多项式的因式分解,粗略地说要遵循如下的步骤:

(1) 在模 P 的整环 Z_P 上分解 $f(x) = F_1F_2\cdots F_m$.若有 $m = 1$,即表明 $f(x)$ 在 $Z_P[x]$ 上不可约,所以在 $Z[x]$ 上也必然不可约.

(2) 根据 $F_1, F_2, \cdots, F_m$ 和它们的组合乘积,不断做提升,看是否可以在 $Z_{P^s}[x]$ 上发现 $f(x)$ 的因子.这个提升过程意味着 s 的不断增大.如果持续这个提升过程而找不到 $f(x)$ 的因子,那么 $f(x)$ 就是不可约的.

关键的一点是当 $f(x)$ 不可约时,希望尽早地结束这个提升过程,这无疑会整体提高分解因式算法的效率.设 B 是 $f(x)$ 的可能的因子的系数的界限.那么当 $P^s > B$ 时,若仍然找不到 $f(x)$ 的因子,就表明 $f(x)$ 是不可分解的.显然,我们希望 B 容易计算且越小越好.

多项式的高 H 是非常简单明了的概念,即 H 是 $f(x)$ 的系数中绝对值最大者. H 是容易算的,其实根本就不必算. 和已知的关于 B 的值比较, H 也是相当小的数. 于是自然希望 H 可以作为多项式 $f(x)$ 因子系数的界限. 遗憾的是,有如下反例:

$$\begin{aligned} f(x) &= x^{14} + x^{12} + x^{10} - x^4 - x^2 - 1 \\ &= (x^7 + 3x^6 + 5x^5 + 6x^4 + 6x^3 + 5x^2 + 3x + 1) \cdot \\ &\quad (x^7 - 3x^6 + 5x^5 - 6x^4 + 6x^3 - 5x^2 + 3x - 1), \end{aligned}$$

H 作为所有因子的界限已不可能. 但是下面的问题是未决的,并且即便是理论上,也极具挑战性.

(1) 若 $f(x)$ 可约,是否可以至少找到一个因子 $g(x)$,使得 H 比 $g(x)$ 的高来得大?

(2) 若 $f(x)$ 可约, $F(x)$ 是 $f(x)$ 的任意的不可约因子、是否 H 比 F 的高大?

这是非常明确的数学问题. 它的解决,看来需要数学家的认真介入. 这个方面,目前最好的结果是找到了 H 可以作为因子系数界限的充分条件,即如果 $f(x)$ 的根不在小半径为 1/2,大半径为 2 的环内出现, H 就可看成是 $f(x)$ 的因子的限界.

另一方面,注意到已知的所有 H 不能作为 $f(x)$ 任意因子的系数界限之反例,它们的根都位于单位圆上. 所以又可以自然地猜测:若 $f(x)$ 的根不在单位圆周上,则 H 就可以作为因子系数的界限. 这个问题若获证实,即表明随机生成的多项式,其高可以被看成是任意因子的系数的界限.

符号计算软件的效率一直是大家所关心的问题. 要想取得更好的效果,必须有深刻的理论做后盾.

参 考 文 献

1 Akritas A G. Elements of computer algebra (with applications). NewYork: John Wiley & Sons, 1989.

2 Buchberger B, Collins G E, Loos R eds. Computer algebra —symbolic and algebraic computation. New York: Springer-Verlag, 1982.

3 Geddes K O, Czapor S R, Labahn G. Algorithms for computer algebra. Boston: Kluwer Academic Publishers, 1992.

4 Knuth D E. The art of Computer programming. Vol. 2: seminumerical algorithms (second edition). Reading, MA: Addison-Wesley, 1981.

5 吴文俊. 几何定理机器证明的基本原理(初等几何部分). 北京:科学出版社, 1984.

6 Hodge W V D, Pedoe D. Methods of algebraic geometry. Vol I. Cambridge, 1953.

7 Franz Winkler. Polynomial Algorithms in Computer Algebra. RISC-LINZ, Austria, 1996.

8 Van der Waerden B L. Modern algebra. Vol Ⅱ. Translated by Theodore J. Benac. New York: Naval Academy, 1950.

9 杨路,张景中,侯晓荣. 非线性代数方程组与定理机器证明. 上海:上海科技教育出版社, 1996.

10 Collins G E . The minimam root seperation of a polynomial. Math of Computation, 1974 (28): 589 ~ 597.

11 Liu Z J . One general criterion for stability. Proceedings of Asian Symposium on Computer Mathematics, Beijing, 1995.

·计算机数学卷·

第18篇

自动定理证明

编　者　李大法
审校者　应明生

目　　录

引　言

定理的自动证明一直是人类的梦想.随着计算机的发展,这一梦想才变得可能.首先王浩教授提出了基于命题逻辑的定理证明系统.然后,鲁宾逊(Robinson)于1965年提出了归结原理,它是基于一阶逻辑的完备的定理证明系统.在此之后,定理证明有了蓬勃的发展.归结原理是一般的,完备的,但不有效.往往由于产生的归结式太多,搜索空间太大,因而消耗了太多的时间和计算机内存而不得不中止证明.为此出现了许多对归结的限制.对归结的限制可能导致破坏完备性,但为提高有效性,也只好忍痛割爱.例如输入归结和单元归结是不完备的,但是很有效.基于归结的定理证明系统已有许多,著名的系统有 Otter 等系统.用 Otter 系统解决了几十年未解决的问题,即鲁宾逊代数是布尔代数,这一成果被列为1997年人工智能五大成就之一.

除了归结方法是完备的外,自然演绎方法和表格方法也是完备的.自然演绎方法因证明步骤可读,证明长度短而受到人们的重视.表格方法容易用树表达,表格方法尤其受到欧洲人的喜欢,每年在欧洲都举行一次有关会议.

归结方法不使用有关领域的知识,不使用有关公理及所证定理的逻辑结构,而导致有效性差.为此人们寻找特定领域的定理证明方法.如吴文俊先生的吴方法,它是基于几何的定理证明方法.重写规则是基于等式的定理证明方法.

1　归结方法

自从20世纪60年代鲁宾逊提出**归结方法**(resolution)以来,出现了许多优化归结的方法.一般化的归结方法不是有效的.归结方法处理的对象是子句公式,因此不得不把一阶逻辑公式转换为子句形式,为此必须把一阶逻辑公式转化成前束范式,然后用斯科伦(Skolem)函数排除存在量词,接着转化为子句形式.

在自然演绎中,有规则MP,按照规则MP,由 $A \to B$ 和 A 可推得 B.归结是MP的一般化.子句是文字的析取.给子句 $l \vee c_1$ 和 $-l \vee c_2$,按照归结可得归结式 $c_1 \vee c_2$,其中 c_1 和 c_2 都是子句.当使用归结方法去证明定理时,总是将所证定理否定,与前提一起化成子句,使用归结,如果推得空子句,则定理成立.

因一般化的归结不是有效的,于是出现了归结的各种提炼.例如语义归结、锁归结、线性归结、输入归结、单元归结.后两者都是不完备的,但对 Horn 集是完备的.由数学定理得到的子句集通常都是 Horn 集.

1.1 前束范式

形式为$(Q_1x_1)(Q_2x_2)\cdots(Q_nx_n)M$的公式为前束范式(prenex normol form),其中Q_i为$\exists$或$\forall$,M中无量词出现.$(Q_1x_1)\cdots(Q_nx_n)$叫前束,M叫矩阵.

例如,$(\forall x)(\exists y)[p(x,y)\rightarrow q(x)]$是前束范式.

1.2 斯科伦函数

例如,公式$F=(\forall x_1)(\forall x_2)\cdots(\forall x_i)(\exists x_{i+1})(Q_{i+2}x_{i+2})\cdots(Q_nx_n)M(x_1,x_2,\cdots,x_n)$是前束范式,用$f(x_1,x_2,\cdots,x_i)$排除存在量词$(\exists x_{i+1})$,其中$f$不出现在$F$中,得到公式$(\forall x_1)(\forall x_2)\cdots(\forall x_i)(Q_{i+2}x_{i+2})\cdots(Q_nx_n)M(x_1x_2,\cdots,x_i,f(x_1,x_2,\cdots,x_i),x_{i+2},\cdots,x_n)$.$f$叫做斯科伦函数.如果存在量词前面无全称量词出现,则用一个新的常量作为斯科伦函数.从左向右依次排除存在量词,排除所有存在量词之后,得到如下公式:$F'=(\forall z_1)(\forall z_2)\cdots(\forall z_l)M'$.此时前束中只有全称量词.不难证明$F$是不可满足的,当且仅当$F'$是不可满足的.由逻辑知识,可知$M'$能转换为CNF形式.往后只考虑$M'$,只要记住$M'$中的变量都是由全称量词所管辖.

例如$(\forall x)(\exists y)(\forall z)[p(x,y)\rightarrow q(y,z)\wedge r(x,z)]$,排除存在量词$(\exists y)$,得到公式$(\forall x)(\forall z)[p(x,f(x))\rightarrow q(f(x),z)\wedge r(x,z)]$,其中$f(x)$是斯科伦函数.将$M'$转换为CNF形式,得到如下公式:$[\neg p(x,f(x))\vee q(f(x),z)]\wedge[\neg p(x,f(x))\vee r(x,z)]$.

1.3 子句及子句集

原子或它的否定叫文字(literal).例如,p、$\neg q$都是文字.零个或几个文字的析取叫子句(clause).含零个文字的子句叫空子句.今后用□表示空子句.例如$p\vee\neg q\vee r$是子句.

排除所有存在量词之后,得到公式$(\forall z_1)(\forall z_2)\cdots(\forall z_l)M$,将$M$转换为CNF形式,即$M=c_1\wedge c_2\wedge\cdots\wedge c_k$,其中$c_i$是子句,即文字的析取.集合$\{c_1,c_2,\cdots,c_k\}$叫子句集.

1.4 基本归结

往下只考虑子句形式公式,即子句.给子句$c_1=p\vee D$和$c_2=\neg p\vee E$,其中p是原子,D和E都是子子句.子句$D\vee E$叫做c_1和c_2的归结式.不难证明$D\vee E$是c_1和c_2的逻辑推论.这种推理规则叫做归结.例如,由子句$\neg p$和p可得到空子句作为归结式.

给定子句集$S=\{c_1,c_2,\cdots,c_n\}$,子句D的演绎是一系列的子句$E_1,E_2,\cdots,E_m$,其中E_m是D,每个E_i是S中的子句或是E_j和E_k的归结式,其中j、$k<i$.如果

由子句集 S 得到空子句的演绎，则称 S 是**不可满足的**.

例如，$S = \{c_1 = p \lor q, c_2 = p \lor \neg q, c_3 = \neg p \lor q, c_4 = \neg p \lor \neg q\}$. 使用归结方法能证明 S 是不可满足的：

$$c_5 = q, \text{Res } c_1 \text{ 和 } c_3,$$
$$c_6 = \neg q, \text{Res } c_2 \text{ 和 } c_4,$$
$$c_7 = \square, \text{Res } c_5 \text{ 和 } c_6.$$

使用归结，由 S 得到空子句作为归结式，因此子句集 S 是不可满足的.

1.5 一阶逻辑的归结

由一阶逻辑公式得到的子句含有变量. 如 $\forall x \exists y[p(x,y) \to q(x,y)]$，可转换为子句 $\neg p(x,f(x)) \lor q(x,f(x))$，变量 x 出现在其中. 怎样把归结引到有变量的子句上去？

例如，$c_1 = q(x,f(y)) \lor r(a,g(x))$，$c_2 = \neg q(b,z) \lor p(z)$. 显然 c_1 与 c_2 中无互补文字对，不能做归结. 在 c_1 中若用 b 替换 x，用 d 替换 y，得 $c'_1 = q(b,f(d)) \lor r(a,g(b))$. 在 c_2 中用 $f(d)$ 替换 z，得 $c'_2 = \neg q(b,f(d)) \lor p(f(d))$. 现在 c'_1 和 c'_2 中有互补文字对，其归结式是 $p(f(d)) \lor r(a,g(b))$. $\theta = \{b/x, d/y, f(d)/z\}$ 叫做**替换**. 因为 θ 统一了 $q(x,f(y))$ 和 $q(b,z)$，因此又叫 θ 为**统一者**(unifier). 写 $c'_1 = c_1\theta$, $c'_2 = c_2\theta$，而且叫 c'_1 和 c'_2 分别为 c_1 和 c_2 的基本例子，因为 c_1' 和 c_2' 中无变量.

不难看出 $q(x,f(y))$ 和 $q(b,z)$ 有许多统一者. 例如 $\alpha = \{b/x, f(y)/z\}$ 也是统一者. 由后面的知识可知，α 是最一般的统一者，简写为 mgu(most general unifier). 在 c_1 中用 b 替换 x，得 $c_1^* = q(b,f(y)) \lor r(a,g(b))$. 在 c_2 中用 $f(y)$ 替换 z 得 $c_2^* = \neg q(b,f(y)) \lor p(f(y))$. c_1^* 和 c_2^* 有互补文字对，可得归结式 $p(f(y)) \lor r(a,g(b))$. 这是最一般的归结式，前面的归结式是后者的例子，只要在后者中用 d 替换 y 就可得到前者.

下面将介绍一般情形下的归结.

1.5.1 替换(substitution)

$\{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$ 叫作替换，其中 t_i 是项，v_i 是变量，$v_i \neq v_j$，$t_i \neq v_i$. 例如，$\{f(a,b)/x, h(c)/y\}$ 是替换. 替换的复合被定义如下：

设
$$\theta = \{t_1/x_1, t_2/x_2, \cdots, t_n/x_n\},$$
$$\lambda = \{u_1/y_1, u_2/y_2, \cdots, u_m/y_m\}$$

都是替换，用 $\theta \circ \lambda$ 表示 θ 和 λ 的复合，考虑集合 $\{t_1\lambda/x_1, \cdots, t_n\lambda/x_n, u_1/y_1, \cdots, u_m/y_m\}$，若 $t_j\lambda = x_j$，则删去 $t_j\lambda/x_j$；若 $y_i \in \{x_1, x_2, \cdots, x_n\}$，则删去 u_i/y_i，剩下的集合为 $\theta \circ \lambda$.

例如 $\theta = \{h(z)/x, w/y\}$，$\lambda = \{d/x, y/w, g(c)/y\}$. $\{h(z)\lambda/x, w\lambda/y, d/x, y/w, g(c)/y\} = \{h(z)/x, y/y, d/x, y/w, g(c)/y\}$，由此集合删去 $y/y, d/x$,

$g(c)/y$,得到集合$\{h(z)/x, y/w\}$,这就是$\theta \circ \lambda$.

替换的复合具有可结合性,即$(\alpha \circ \beta) \circ r = \alpha \circ (\beta \circ r)$,其中$\alpha$、$\beta$和$r$都是替换.但替换的复合是不可交换的,即$\alpha \circ \beta \neq \beta \circ \alpha$.

设E是表达式,θ是替换.怎样求$E\theta$?令$\theta = \{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$,在$E$中同时用$t_i$替换$v_i$的每一次出现,$i = 1,2,\cdots,n$,得到的表达式就是$E\theta$.$E\theta$叫作$E$的一个例子.

例如$c_1 = p(y) \vee q(x)$,$\sigma = \{f(x)/y, y/x\}$,则$c_1\sigma$应该是$p(f(x)) \vee q(y)$.

下面的做法是错误的,先对y做替换,得$p(f(x)) \vee q(x)$,再对x做替换,得$p(f(y)) \vee q(y)$,这违背了同时替换的原则.

不难证明

$$(\alpha \circ \beta) \circ r = \alpha \circ (\beta \circ r).$$

通常

$$\alpha \circ \beta \neq \beta \circ \alpha.$$

1.5.2 统一算法(unification algorithm)

令E_1、E_2,$\cdots$,E_k是表达式,θ是替换.如果$E_1\theta = E_2\theta = \cdots = E_k\theta$,则$\theta$叫做$E_1, E_2, \cdots, E_k$的统一者,称$E_1, E_2, \cdots, E_k$是可统一的.设$\sigma$是$E_1, E_2, \cdots, E_k$的统一者.如对其每一个统一者$\theta$,都有替换$\lambda$使得$\theta = \sigma \circ \lambda$,则叫$\sigma$为$E_1, E_2, \cdots, E_k$的mgu(most general unifier).

例如,$E_1 = p(a, x)$,$E_2 = p(a, f(y))$,$\theta = \{f(b)/x, b/y\}$,$E_1\theta = E_2\theta = p(a, f(b))$,因此$\theta$是$E_1$和$E_2$的统一者.

再如,$\sigma = \{f(y)/x\}$,$E_1\sigma = E_2\sigma = p(a, f(y))$,$\sigma$也是统一者,而且$\sigma$是$E_1$和$E_2$的mgu.例如$\theta = \sigma \circ \lambda$,其中$\lambda = \{b/y\}$.

为了介绍统一算法,还要介绍什么是不一致集合.例如,$E_1 = p(a, x)$,$E_2 = p(a, f(y))$,则E_1和E_2的不一致集合是$\{x, f(y)\}$.用D表示不一致集合.关于不一致集合的定义见参考文献[1].

现在可以介绍统一算法.

步1　置$k = 0$,$S_k = S$和$\sigma_k = \varepsilon$,其中S是输入公式集,ε是空替换.

步2　如果S_k是单元素集,则停,σ_k是S的MGU,否则寻找S_k的不一致集D_k.

步3　如果D_k中有变量v_k和项t_k使得v_k不出现在t_k之中,则$\sigma_{k+1} = \sigma_k \circ \{t_k/v_k\}$和$S_{k+1} = S_k \circ \{t_k/v_k\}$,否则停,$S$是不可统一的.

步4　置$k = k + 1$,且转向步2.

例1　求$p(f(a), g(x))$和$p(y, y)$的mgu.

令$S = \{p(f(a), g(x)), p(y, y)\}$.下面的步$i$是指统一算法中的步$i$.

步1　$k = 0$,$\sigma_0 = \varepsilon$.

步2　$S\sigma_0 = S$,$D_0 = \{f(a), y\}$.

步3　$\sigma_1 = \sigma_0 \circ \{f(a)/y\} = \{f(a)/y\}$,置$k = 1$.

步2　$S\sigma_1 = \{p(f(a), g(x)), p(f(a), f(a))\}$,
$D_1 = \{g(x), f(a)\}$.

步 3 $p(f(a),g(x))$ 和 $p(y,y)$ 是不可统一的.

例 2 求 $S=\{p(a,x,h(g(z))),p(z,h(y),h(y))\}$ 的 mgu.

下面的步 i 是指统一算法中的步 i.

步 1 $k=0,\sigma_0=\varepsilon$.

步 2 $S\sigma_0=S,D_0=\{a,z\}$.

步 3 $\sigma_1=\{a/z\}$,置 $k=1$.

步 2 $S\sigma_1=\{p(a,x,h(g(a))),p(a,h(y),h(y))\}$,
$D_1=\{h(y),x\}$.

步 3 $\sigma_2=\sigma_1\circ\{h(y)/x\}=\{a/z,h(y)/x\}$,置 $k=2$.

步 2 $S\sigma_2=\{p(a,h(y),h(g(a))),p(a,h(y),h(y))\}$,
$D_2=\{g(a),y\}$.

步 3 $\sigma_3=\sigma_2\circ\{g(a)/y\}=\{a/z,h(g(a))/x,g(a)/y\}$,置 $k=3$.

步 2 $S\sigma_3=\{p(a,h(g(a)),h(g(a)))\}$,$\sigma_3$ 是 mgu.

定理 1 (统一定理) 如果 W 是非空可统一者的表达式集,则统一算法总是停在步骤 2 而且最后的 σ_k 是 W 的最一般统一者.

1.5.3 二元归结(binary resolution)

令 c_1 和 c_2 是两个没有共同变量的子句. L_1 和 L_2 分别是 c_1 和 c_2 的文字. 如果 L_1 和 $\neg L_2$ 有 mgu σ,则$(c_1\sigma-L_1\sigma)\cup(c_2\sigma-L_2\sigma)$ 是 c_1 和 c_2 的**二元归结式**. 在这里子句被看作文字的集合,L_1 和 L_2 叫作被归结文字,c_1 和 c_2 叫作父母子句.

例如,$c_1=r(x,a)\vee\neg q(b),c_2=\neg r(f(b),y)\vee\neg p(h(y)),L_1=r(x,a),L_2=\neg r(f(b),y),\sigma=\{f(b)/x,a/y\}$,则$\neg q(b)\vee\neg p(h(a))$ 是 c_1 和 c_2 的二元归结式.

1.5.4 因子(factor)

例如,子句 $c=\neg p(x)\vee\neg p(y)\vee\neg p(f(a))\vee q(y)$,显然 $\neg p(x)$、$\neg p(y)$、$\neg p(f(a))$ 有 mgu $\sigma=\{f(a)/x,f(a)/y\}$,$c\sigma=\neg p(f(a))\vee q(f(a))$,$c\sigma$ 叫作c的**因子**.

1.5.5 归结

二元归结是不完备的,即是说从不可满足的子句集可能推导不出空子句. 例如,$c_1=p(x)\vee p(y),c_2=\neg p(u)\vee\neg p(v)$,$\{c_1,c_2\}$ 是不可满足的,但是使用二元归结不能从 c_1 和 c_2 推导空子句. $p(x)$ 是 $p(x)\vee p(y)$ 的因子,$\neg p(u)$ 是 $\neg p(u)\vee\neg p(v)$ 的因子,由 $p(x)$ 和 $\neg p(u)$ 可推导出空子句.

c_1(或 c_1 的因子) 和 c_2(或 c_2 的因子) 的二元归结式叫归结式.

定理 2 子句集 S 是不可满足的充要条件是使用归结由 S 可推导出空子句.

例 3 苏格拉底断言:"人都是要死的,苏格拉底是人. 因此苏格拉底也是要死的."用 Man(x) 表示 x 是人. 显然 Man(s) 表示苏格拉底是人. Mortal(x) 表示 x 是

要死的."人都是要死的"形式化为"$\forall x[\mathrm{Man}(x)\rightarrow\mathrm{Mortal}(x)]$". $\mathrm{Mortal}(s)$ 表示苏格拉底是要死的.将结论否定,得子句 $c_1=\neg\mathrm{Mortal}(s)$.从前提可得子句 $c_2=\neg\mathrm{Man}(x)\vee\mathrm{Mortal}(x)$, $c_3=\mathrm{Man}(s)$.现在使用归结证明子句集 $\{c_1,c_2,c_3\}$ 是不可满足的.

$c_4=\mathrm{Mortal}(s)$ Res c_2 和 c_3

$c_5=\square$ Res c_4 和 c_1

根据定理子句集 $\{c_1,c_2,c_3\}$ 是不可满足的.因此 $c_2\wedge c_3\Rightarrow c_1$.

归结简单、易编程,但有效性差.使用水平饱和方法,从 $p\vee q$, $\neg p\vee q$, $p\vee\neg q$, $\neg p\vee\neg q$ 推出空子句需要 34 个归结.为了提高归结的有效性,出现了归结的各种提炼(refinement)和各种删除策略.

1.5.6 语义归结(semantic resolution)

前面已定义了归结,有两个问题需要解决.一是子句集 S 有 n 个子句,哪两个子句做归结,归结原理中并未解决.简单的办法就是组合式归结,两两做归结.当有新归结式产生时,新归结式与 S 的子句及其它归结式分别做归结.显然要产生组合爆炸.二是 c_1 和 c_2 做归结时,并未指明哪一对文字是被归结文字,因为 c_1 和 c_2 中可能有多对互补文字对,当然也就有多种选择,搜索树也就相应变大.这一节就是讨论这两个问题.

关于问题 1,为了阻止组合爆炸,把子句集 S 分为两个子集,使得同一个子集内不做归结.问题是依什么标准来分.用解释 I 来分,凡是在 I 上为真的子句为一个子集,在 I 上为假的子句为另一个子集.

例如,$S=\{(1)\neg p\vee\neg q\vee r,(2)p\vee r,(3)q\vee r,(4)\neg r\}$,取 $I=\{\neg p,\neg q,\neg r\}$, $S_1=\{p\vee r,q\vee r\}$,其中 $p\vee r$、$q\vee r$ 在 I 上为假;$S_2=\{\neg p\vee\neg q\vee r,\neg r\}$,其中 $\neg p\vee\neg q\vee r$ 和 $\neg r$ 在 I 上为真. S_1 里两子句不做归结,S_2 里两子句也不做归结. S_1 的子句可以分别与 S_2 的子句做归结.这样大大减少了归结数量.下面是所做的归结.

(5) $\neg q\vee r$ Res(1) 和(2)

(6) $\neg p\vee r$ Res(1) 和(3)

(7) p Res(2) 和(4)

(8) q Res(3) 和(4)

阻止了子句(1) 和(4) 做归结.

(9) r Res(5) 和(8)

(10) $\square$ Res(9) 和(4)

关于问题 2,给一对子句 c_1 和 c_2,可能有多对互补文字对,加上某种限制使得只有一对互补文字对做归结.这里使用谓词符号次序.例如 $p>q>r$.当做归结 $\mathrm{Res}(c_1,c_2)$ 时,例如规定 c_1 中具有最大谓词符号次序的文字选做被归结文字.在上例中,规定 $p>q>r$,在此条件下 $p\vee r$ 和 $\neg r$ 不能做归结,因为 p 具有最大谓词符号次序.如此,上例中归结式(7) 和(8) 将不会出现.

下面定义语义归结.

令 I 是一个解释，而 p 是谓词符号次序. $\{E_1,E_2,\cdots,E_q,N\}$ 叫做**语义碰撞**(semantic clash)，当且仅当下面的条件被满足：

1° $E_i, i = 1,2,\cdots,q$，在 I 上为假.

2° 令 $R_1 = N, R_{i+1}$ 是 R_i 和 E_i 的归结式，$i = 1,2,\cdots,q$，其中 R_{q+1} 在 I 上为假，而且 E_i 中被归结的文字含有 E_i 中的最大谓词符号，R_{q+1} 叫作 ***pI* 归结式**.

例如 $E_1 = p, E_2 = q, E_3 = r, N = s \vee \neg p \vee \neg q \vee \neg r, I = \{\neg p, \neg q, \neg r, \neg s\}$，$p > q > r > s$，$I$ 是解释，s 是 pI 归结式.

S 是一个子句集，I 是解释而 P 是出现在 S 中的谓词次序. 子句序列 $c_1, c_2, \cdots, c_n$ 叫做 ***pI* 演绎**，当且仅当每个 c_i 是 S 中的子句或是 pI 归结式.

1.5.7　超归结(hyperresolution)

在语义归结中使用解释 I 将子句集分为两个子集，使得同一子集的子句不做归结，以此提高归结的有效性. 这又产生了新的问题，通常 Herbrand 解释是无穷的，怎样指定解释呢?为了简单起见，通常取解释的成员都是正文字或都是负文字，即 $I = \{m_1, m_2, m_3, \cdots\}$ 或 $I = \{-m_1, -m_2, \cdots\}$，其中 m_i 是原子.

在 pI 归结中，如 I 中每一个文字都是负文字，此时所有 E_i 和 E_{q+1} 都是正的，叫作**正超归结**.

在 pI 归结中，如 I 中每一个文字都是正的，此时所有 E_i 和 E_{q+1} 都是负的，叫作**负超归结**.

1.5.8　锁归结(lock resolution)

对锁归结来说，每个子句中的每个文字都用整数做标志. 例如，${}_5q(x) \vee {}_7\neg p(x)$.

c 是一子句，如 c 中有两个或更多的文字有 mgu σ，在 $c\sigma$ 中对同样的文字只保留具有最小标志的文字. 如此由 $c\sigma$ 得到的子句叫做 c 的**锁因子**. 例如，$c = {}_5q(x) \vee {}_{10}q(f(a)) \vee {}_7\neg p(h(x))$，$\sigma = \{f(a)/x\}$，$c\sigma = {}_5q(f(a)) \vee {}_{10}q(f(a)) \vee {}_7\neg p(h(f(a)))$，${}_5q(f(a)) \vee {}_7\neg p(h(f(a)))$ 是 c 的锁因子.

c_1 和 c_2 是两个无公共变量的子句，c_1 和 c_2 中每个文字都加标志. L_1 和 L_2 分别是 c_1 和 c_2 中具有最小标志的文字. 如果 L_1 和 $\neg L_2$ 有 mgu σ，则在 $c_1\sigma \cup c_2\sigma$ 中去掉 $L_1\sigma$ 和 $L_2\sigma$，而且对同样的文字只保留最小标志，如此得到的子句叫作**二元锁归结**. 在这里子句看做文字集合.

子句 c_1(或 c_1 的锁因子)和 c_2(或 c_2 的锁因子)的二元锁归结式叫做锁归结式.

例如 $c_1 = {}_3p(x, f(y)) \vee {}_5r(h(x)) \vee {}_4\neg q(x)$，$c_2 = {}_{10}\neg q(a) \vee {}_8\neg p(a, f(b))$，${}_3p(x, f(y))$ 和 ${}_8p(a, f(b))$ 有 mgu$\sigma = \{a/x, b/y\}$，从 $c_1\sigma \vee c_2\sigma$ 中删去 ${}_3p(x, f(y))\sigma$ 和 ${}_8\neg p(a, f(b))\sigma$，得 ${}_5r(h(a)) \vee {}_4\neg q(a) \vee {}_{10}\neg q(a)$，然后对同样的文字只保留最小标志文字，得二元锁归结式 ${}_5r(h(a)) \vee {}_4\neg q(a)$.

锁演绎被定义如下：S 是子句集，其中每个文字都用正整除做标志，子句序列 $c_1, c_2, \cdots, c_n$ 叫做锁演绎，如果每个 c_i 或是 S 的子句或是前面子句的锁归结式.

例 4　设 $S = \{(1)_1 r, (2)_2 q \vee {}_3\neg r, (3)_4 s \vee {}_5\neg r, (4)_6 p \vee {}_7\neg q \vee {}_8\neg s, (5)_9\neg p \vee {}_{10}\neg q \vee {}_{11}\neg s\}$，利用锁归结证明它是不可满足的.

证明

$(6)_7\neg q \vee {}_8\neg s$	Res(4) 和(5)
$(7)_3\neg r \vee {}_8\neg s$	Res(6) 和(2)
$(8)_8\neg s$	Res(7) 和(1)
$(9)_5\neg r$	Res(8) 和(3)
(10) □	Res(9) 和(1)

1.5.9　线性归结(linear resolution)

S 是子句集，子句序列 $c_0, c_1, c_2, \cdots, c_n$ 叫做**线性演绎**，如果 $c_0 \in S$，c_{i+1} 是 c_i 和 B_i 的归结式，$B_i \in S$ 或 B_i 是 c_j，$j < i$，c_i **叫做中心子句**，B_i 叫作**边子句**，c_0 叫作**顶子句**.

线性归结可以用图 1-1 描述.

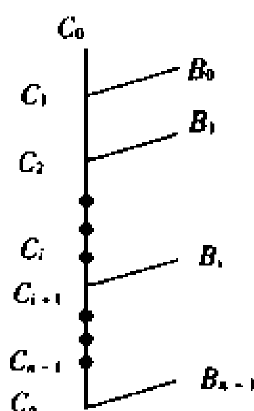


图 1-1

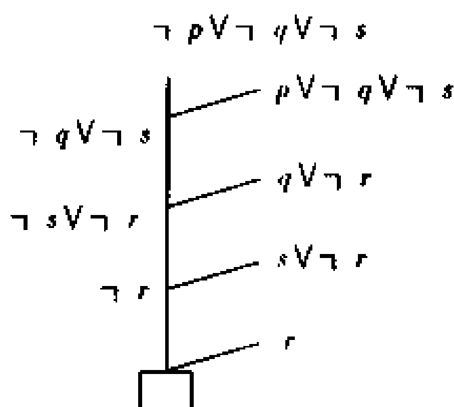


图 1-2

例如子句集 $S = \{r, q \vee \neg r, s \vee \neg r, p \vee \neg q \vee \neg s, \neg p \vee \neg q \vee \neg s\}$，用线性归结可证明它是不可满足的，如图 1-2.

1.5.10　输入归结(input resolution)

输入归结是一个特殊的线性归结，其中每个边子句都是 S 的子句.

输入归结显然比线性归结有效，遗憾的是它是不完备的，但对 Horn 集是完备的. prolog 的实现使用的正是输入归结.

Horn 集被定义如下，一个子句叫做 Horn 子句，如果它至多含有一个正文字；一个子句集叫 Horn 集，如果其每一个子句都是 Horn 子句.

上例中的 $S = \{r, q \vee \neg r, s \vee \neg r, p \vee \neg q \vee \neg s, \neg p \vee \neg q \vee \neg s\}$ 是 Horn 集，因此 S 有一个输入归结证明，上例给出的线性证明恰是输入归结证明.

不仅输入归结对 Horn 集是完备的，而且输入二元归结，即不使用因子规则，对 Horn 集也是完备的.

1.5.11　单元归结(unit resolution)

一个子句如果只包含一个文字,则该子句叫单元子句,如果单元子句含的是正文字叫正单元子句,如果单元子句含的是负文字叫负单元子句.

子句 c_1 和 c_2 做归结时,若限制其中的一个为单元子句,则该归结叫**单元归结**.若限制其中的一个为正单元子句,则该归结叫正单元归结.

单元归结是有效的,但是不完备.然而对 Horn 集是完备的.正单元归结对 Horn 集也是完备的.不仅如此,二元正单元归结对 Horn 集也是完备的.

例如线性归结小节所举的例,$S = \{(1)r,(2)q \vee \neg r,(3)s \vee \neg r,(4)p \vee \neg q \vee \neg s,(5)\neg p \vee \neg q \vee \neg s\}$,$S$ 是 Horn 集.它有正单元归结证明如下:

(6)q	Res(1) 和(2)
(7)s	Res(1) 和(3)
(8)$p \vee \neg q$	Res(4) 和(7)
(9)p	Res(8) 和(6)
(10)$\neg p \vee \neg q$	Res(5) 和(7)
(11)$\neg p$	Res(10) 和(6)
(12)$\square$	Res(9) 和(11)

2　自然演绎

自然演绎系统由 Gentzen 系统改写而来.它允许使用五个连结词 $\rightarrow$, $\vee$, $\wedge$, $\neg$, $\leftrightarrow$ 和两个量词 $\forall$ 和 $\exists$.自然演绎使用的推理规则都是很自然而且容易接受的.如 MP 规则正是人们常说的三段论式,即大前提 $A \rightarrow B$,小前提 A 和结论 B.因此定理的自然演绎证明是自然的而且是结构化的,可读的.

归结方法证明与自然演绎证明区别在于:对归结证明来说,先将所证定理否定,与前提一起斯科伦化,再将公式化为子句形式,若推导出空子句,即所证定理为真.对自然演绎来说,对公式不做任何转换,即保留原公式结构,由前提推导结论.由于归结引进斯科伦函数,由此可能产生许多无关的归结式,造成组合爆炸.而自然演绎排除存在量词时引进的是常量而不是函数,搜索空间较小,证明长度比起归结要短.对自然演绎来说,合式公式的子公式性质可用上,增强了有效性.自然演绎推理充分利用公式的结构,首先从结论向前提方向推导,再由前提向结论方向推导.如定理具有 $A \rightarrow B$ 形式,则假定 A 推导B;若定理具有 $A \wedge B$形式,则产生两个新子目标,分别证 A 和B,若证得 A 和B,则 $A \wedge B$ 也成立;若所证定理具有 $A \leftrightarrow B$ 形式,则转去证明$[A \rightarrow B] \wedge [B \rightarrow A]$;若所证定理具有形式 $A \vee B$,则只要证 $\sim A \rightarrow B$;若所证定理具有形式 $\forall x B(x)$,则只要证 $B(x)$.显然这种推理方式是很自然的,也是人们所熟悉的.由下往上证明,其路径是唯一的.由上往下推理时,尽量使用命题逻辑推理规则,不得已时使用关于量词的规则.为了尽量使用命题逻辑推理规则,求量词最小化是必要的,即尽量把量词往里推.如 $\forall x \exists y\ [p(x) \rightarrow$

$q(y)]$,量词最小化后得公式 $\exists xp(x)\rightarrow\exists yq(y)$,对后者命题逻辑规则可能用上.

自然演绎由 Gentzen 系统改写而来,它有许多推理规则.自然演绎格式是 $\Gamma\vdash A$,其中 Γ 是前提集合,A 是公式.下面将逐一介绍这些规则.

1. 基本公理

(1) $\Gamma, A\vdash A$.

(2) $\dfrac{\Gamma\vdash B}{\Gamma, A\vdash B}$.

2. 关于命题逻辑推理规则

(1) $\dfrac{\Gamma\vdash A\quad \Gamma\vdash A\rightarrow B}{\Gamma\vdash B}$, MP.

(2) $\dfrac{\Gamma\vdash \neg B\quad \Gamma\vdash A\rightarrow B}{\Gamma\vdash \neg A}$, MT.

(3) $\dfrac{\Gamma\vdash A\vee B\quad \Gamma\vdash \neg A}{\Gamma\vdash B}$, DS.

(4) $\dfrac{\Gamma\vdash A\vee B\quad \Gamma\vdash \neg B}{\Gamma\vdash A}$, DS.

(5) $\dfrac{\Gamma, A\vdash B}{\Gamma\vdash A\rightarrow B}$, CP.

(6) $\dfrac{\Gamma\vdash A}{\Gamma\vdash A\vee B}$, $\dfrac{\Gamma\vdash B}{\Gamma\vdash A\vee B}$, ADD.

(7) $\dfrac{\Gamma\vdash A\quad \Gamma\vdash B}{\Gamma\vdash A\wedge B}$, Conjunction.

(8) $\dfrac{\Gamma\vdash A\wedge B}{\Gamma\vdash A}$, $\dfrac{\Gamma\vdash A\wedge B}{\Gamma\vdash B}$, Simplication.

(9) $\dfrac{\Gamma\vdash A\quad \Gamma\vdash \neg A}{\Gamma\vdash B}$, $\neg$ elimination.

(10) $\dfrac{\Gamma, \neg A\vdash B\quad \Gamma, \neg A\vdash \neg B}{\Gamma\vdash A}$, IP.

(11) $\dfrac{\Gamma\vdash A\rightarrow B\quad \Gamma\vdash B\rightarrow A}{\Gamma\vdash A\leftrightarrow B}$, $\dfrac{\Gamma\vdash A\leftrightarrow B}{\Gamma\vdash [A\rightarrow B]\wedge [B\rightarrow A]}$, Equivalence.

(12) $\dfrac{\Gamma\vdash A\vee B}{\Gamma\vdash \neg A\rightarrow B}$, $\dfrac{\Gamma\vdash \neg A\rightarrow B}{\Gamma\vdash A\vee B}$, IMP.

(13) $\dfrac{\Gamma\vdash \neg\neg A}{\Gamma\vdash A}$, Double negation.

(14) $\dfrac{\Gamma\vdash \neg [A\vee B]}{\Gamma\vdash \neg A\wedge \neg B}$, $\dfrac{\Gamma\vdash \neg [A\wedge B]}{\Gamma\vdash \neg A\vee \neg B}$, DeMorgan,

3. 关于量词的规则

(1) $\dfrac{\Gamma\vdash A(x)}{\Gamma\vdash \forall xA(x)}$, UG.

其中 x 不在 Γ 中自由出现.

(2) $\dfrac{\Gamma\vdash A(t)}{\Gamma\vdash (\exists x)A(x)}$, EG.

其中 t 在 $A(x)$ 中对 x 是自由的.

(3) $\dfrac{\Gamma \vdash (\forall x)A(x)}{\Gamma \vdash A(t)}$, US.

其中 t 在 $A(x)$ 中对 x 是自由的.

(4) $\dfrac{\Gamma \vdash (\exists x)A(x),\ \Gamma, A(a) \vdash c}{\Gamma \vdash c}$, ES.

其中 a 是常量,且不在 $\Gamma,(\exists x)A(x)$ 或 c 中出现.

(5) $\dfrac{\Gamma \vdash \neg(\forall x)A(x)}{\Gamma \vdash (\exists x)\neg A(x)}, \dfrac{\Gamma \vdash \neg(\exists x)A(x)}{\Gamma \vdash (\forall x)\neg A(x)}$.

例如,苏格拉底断言被形式化如下:

人都是要死的,形式化为 $\forall x\ [\mathrm{Man}(x) \to \mathrm{Mortal}(x)]$;苏格拉底是人,形式化为 $\mathrm{Man}(s)$;苏格拉底是要死的,形式化为 $\mathrm{Mortal}(s)$. 由前提人都是要死的和苏格拉底是人推导苏格拉底是要死的. 使用自然演绎也能证明它,在此之前曾用归结方法证明过它.

(1) $\{1\} \vdash \forall x\ [\mathrm{Man}(x) \to \mathrm{Mortal}(x)]$ premise.
(2) $\{1,2\} \vdash \mathrm{Man}(s)$ premise.
(3) $\{1,2\} \vdash \mathrm{Man}(s) \to \mathrm{Mortal}(s)$ US1.
(4) $\{1,2\} \vdash \mathrm{Mortal}(s)$ MP2,3

3 表 格 法

符号公式:A 是公式,F_A 表示 A 是假,T_A 表示 A 是真.

表格是一棵树,其结点带有符号公式.

F_A 和 T_A 称配对公式. 如果一个分支含有配对公式 T_A 和 F_A,则该分枝叫做闭分枝. 一个表格是闭的,如果它的所有分支都是闭的.

$\vdash A$ 表示 A 有一个证明. A 的一个证明是 F_A 的闭表格.

关于分析表格的推理规则:

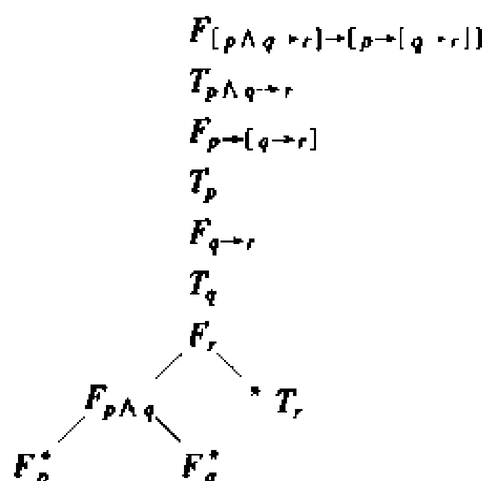
$$(\neg)\ \frac{T_{\neg A}}{FA},\quad \frac{F_{\neg A}}{TA}$$

$$(\wedge)\ \frac{T_{A \wedge B}}{T_A, T_B}$$

$$(\vee)\ \frac{T_{A \vee B}}{T_A \mid T_B}, \frac{F_{A \vee B}}{F_A, F_B}$$

$$(\to)\ \frac{T_{A \to B}}{F_A \mid T_B}, \frac{F_{A \to B}}{T_A, F_B}$$

例如,$\vdash [p \wedge q \to r] \to [p \to [q \to r]]$,用分析表格法证明它:



4　关于带等词的定理证明

4.1　关于等号公理

令 S 是子句集,关于 S 的等号公理如下:

(1)$x = x$.　　反身公理

(2)$x \neq y \vee y = x$.　　对称公理

(3)$x \neq y \vee y \neq z \vee x = z$.　　传递公理

(4)$x_j \neq x_0 \vee \neg p(x_1, \cdots, x_j, \cdots, x_n) \vee p(x_1, \cdots, x_0, \cdots, x_n)$, $j = 1, 2, \cdots, n$, p 是出现在 S 中的任何谓词符号.　　关于谓词的替换公理

(5)$x_j \neq x_0 \vee f(x_1, \cdots, x_j, \cdots, x_n) = f(x_1, \cdots, x_0, \cdots, x_n)$, $j = 1, 2, \cdots, n$, f 是出现在 S 中的任何函数符号.　　关于函数的替换公理

定理 1　设 S 是子句集,K 是关于 S 的等号公理集. S 是 E 不可满足,当且仅当 $(S \cup K)$ 是不可满足.

关于 E 不可满足的定义请见参考文献[1].

4.2　调　　换

c_1 和 c_2 是两个无公共变量的子句(叫做父母句),如果 c_1 是 $L[t] \vee c'_1$, c_2 是 $r = s \vee c'_2$,其中 $L[t]$ 是一个含项 t 的文字,而 c'_1 和 c'_2 都是子句,并且 t 和 r 有 mgu σ,则 $L\sigma[s\sigma] \vee c'_1\sigma \vee c'_2\sigma$ 叫做 c_1 和 c_2 的**二元调换式**. 其中 $L\sigma[s\sigma]$ 表示在 $L\sigma$ 中用 $s\sigma$ 替换 $t\sigma$ 的一次出现而得到的结果,文字 L 和 $r = s$ 叫做被调换的文字.

c_1(或 c_1 的因子)和 c_2(或 c_2 的因子)的二元调换式叫做调换式.

例如,$p(f(x)) \vee q(h(x))$, $a = b \vee r(c)$, $\sigma = \{a/x\}$,二元调换式是 $p(f(b)) \vee q(h(a)) \vee r(c)$.

4.3　超调换与输入调换

P 是谓词符号次序，c_1 和 c_2 的调换式叫做 P 超调换(hyperparamodulation)，当且仅当 c_1 和 c_2 都是正子句且 c_1 和 c_2 中被调换的文字都含有 c_1 和 c_2 的最大的谓词符号.

如果一个调换的父母句之一是输入子句，则该调换叫输入调换.

例如，$S = \{a = b, c = d, f(b) = g(c), f(a) \neq g(d)\}$，$S$ 有如图 4-1 所示的输入调换反驳：

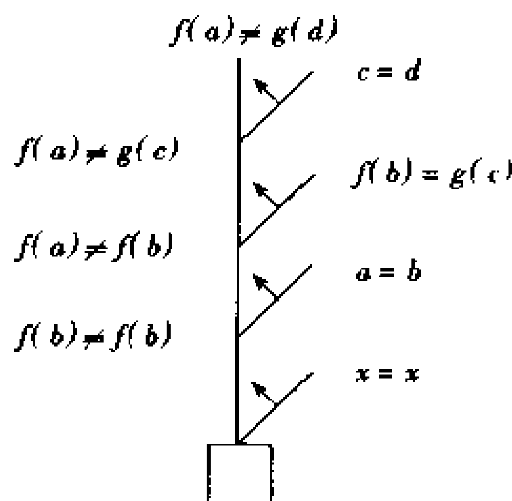


图 4-1

上面的输入调换反驳既是单元调换反驳，也是线性调换反驳.

4.4　单元调换与线性调换

如果一个调换的父母句之一是单元子句或是一个子句的单元因子，则该调换为单元调换.

S 是子句集. 子句序列 $c_0, c_1, c_2, \cdots, c_n$ 叫做关于调换和归结的线性演绎，如果 c_{i+1} 是 c_i 和 B_i 的调换式或归结式，其中 $c_0 \in S$, $B_i \in S$ 或 B_i 是 c_j, $j < i$.

5　重 写 规 则

Knuth 和 Bendix 于 1970 年提出重写规则. 其想法是给出等式集 $E = \{t_1 = s_1, t_2 = s_2, \cdots, t_n = s_n\}$，问 $t = s$ 是否能由上述等式推导出来. 先把 E 中的等式看作化简，如等式 $\alpha = \beta$，把 β 看作比 α 更简单，于是把 $\alpha \to \beta$ 看作化简规则. 因此从等式公理集 E 可得重写规则集 $\{t_1 \to s_1, t_2 \to s_2, \cdots, t_n \to s_n\}$. 有了重写规则集，任给一词都可一步步地化简它. 如群论中 $a^{-1}(ab) = b$ 可看作化简规则 $a^{-1}(ab) \to b$. 任何公式具有形式 $a^{-1}(ab)$ 都可化简到 b. 通常如词 α 有形式 $\varphi\beta\psi$，β 是 α 的子词，若有化

简规则 $\lambda \to \sigma$ 使得 $\beta = \lambda\theta$，其中 θ 是替换，则有 $\beta' = \sigma\theta$，而且说 α 化简到 $\alpha' = \varphi\beta'\gamma$，写 $\alpha \to \alpha'$. 例如用 $a^{-1}(ab) \to b$ 可化简 $((x^{-1})^{-1}(x^{-1}(xy)))$ 到 xy 和 $(x^{-1})^{-1}y$. 如果对每一个词 t 都有唯一的不可化简的词 t^*，则判断 $t = s$ 是否成立问题就转向把为 t 和 s 化简到不可化简的形式，然后再判断其最简形式是否相等.

给定关系 $R = \{(\lambda_1,\sigma_1),(\lambda_2,\sigma_2),\cdots,(\lambda_m,\sigma_m)\}$，$(\lambda_i,\sigma_i)$ 看作词对. α 是词，β 是 α 的子词，$\alpha = \varphi\beta\psi$，$\varphi$ 和 ψ 是符号串. 假如 (λ,σ) 属于 R 使得 $\beta = \lambda\{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$，令 $\beta' = \sigma\{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$，$\alpha' = \alpha\beta'\psi$，就说在 R 下 α 简化到 α'，写作 $\alpha \to \alpha'(R)$，其中 $\{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$ 是替换.

例如，群可由下面三个公理来定义：① $e \cdot x = x$，e 是左单位元素，② $x^{-1} \cdot x = e$，任何元素都有左逆，③ $(x \cdot y) \cdot z = x \cdot (y \cdot z)$，关于运算"·"结合律成立. 由此可得到重写关系 $R\{(x^{-1} \cdot x, e), (e \cdot x, x), ((x \cdot y) \cdot z, x \cdot (y \cdot z))\}$.

例如词 $(a^{-1} \cdot a) \cdot b$，子词有 $a^{-1}, a, b, (a^{-1} \cdot a), (a^{-1} \cdot a) \cdot b$，利用 R 可以重写 $(a^{-1} \cdot a) \cdot b$ 为 $e \cdot b$. 令 β 是子词 $(a^{-1} \cdot a)$，显然 $\beta = (x^{-1} \cdot x)\{a/x\}$，$\{a/x\}$ 是替换，$\beta' = e\{a/x\} = e$，注意 e 是常量. $\alpha' = e \cdot b$.

"$\to$"定义为简化关系，"$\stackrel{*}{\to}$"定义为"$\to$"的反身、传递闭包. $a \stackrel{*}{\to} b$ 意味着对序列 $a_0, a_1, \cdots, a_n (n \geqslant 0)$，使得 $a = a_0, b = a_n$，对于 $0 \leqslant j < n$，有 $a_j \to a_{j+1}(R)$. 例如 $(a^{-1}a) \cdot b$ 可简化为 $e \cdot b$，写为 $(a^{-1} \cdot a) \cdot b \to e \cdot b$，利用 R 中的 $(e \cdot x, x)$ 可将 $e \cdot b$ 重写为 b. 因为 $e \cdot b = (e \cdot x)\{b/x\}$，$\{b/x\}$ 是替换，最后 $e \cdot b \to b$. 所以 $(a^{-1} \cdot a) \cdot b \to e \cdot b \to b$，可写为 $(a^{-1} \cdot a) \cdot b \stackrel{*}{\to} b$.

"$\stackrel{*}{\leftrightarrow}$"定义为关系"$\to$"的反身、对称、传递闭包，即"$\stackrel{*}{\leftrightarrow}$"是等价关系. $a \stackrel{*}{\leftrightarrow} b$ 意味着 $a \stackrel{*}{\to} b$ 或 $b \stackrel{*}{\to} a$.

给定词 α 和 β 以及关系 R，词的问题是决定是否有 $\alpha \stackrel{*}{\leftrightarrow} \beta(R)$. 例如，在上面的例子中 $R = \{(e \cdot x, x), (x^{-1} \cdot x, e), ((x \cdot y) \cdot z, x \cdot (y \cdot z))\}$，给定 $b \cdot e$ 和 b，问 $b \cdot e \stackrel{*}{\leftrightarrow} b$ 成立吗?

1. 不可化简元素

词 α 关于关系 R 不可化简，如果没有 α' 使得 $\alpha \to \alpha'$.

如上例的 R，词 e 关于 R 是不可化简的.

2. 终止关系(termination)

如每个系列 $a_0 \to a_1 \to a_2 \to \cdots$ 都终止，则关系 R 是终止的.

3. 合流(confluence)

如果 $a \stackrel{*}{\to} a'$ 而且 $a \stackrel{*}{\to} a''$，则有 b 使得 $a' \stackrel{*}{\to} b$ 而且 $a'' \stackrel{*}{\to} b$，如图 5-1 所示，则说 R 是合流的.

4. Church-Rosser 性质

如果 $a \stackrel{*}{\leftrightarrow} b$，则有 c 使得 $a \stackrel{*}{\to} c$ 而 $b \stackrel{*}{\to} c$，如图 5-2 所示，则说 R 有 Church-Rosser 性质.

5. 局部合流(local confluence)

如果 $a \to a'$ 而且 $a \to a''$，则有 b 使得 $a' \stackrel{*}{\to} b$ 而且 $a'' \stackrel{*}{\to} b$，如图 5-3 所示，那

么 R 是局部合流.

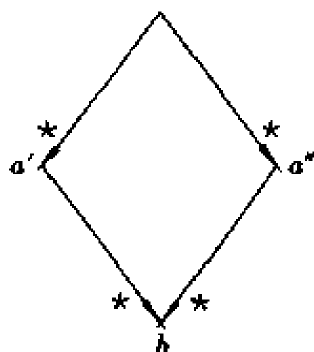


图 5-1

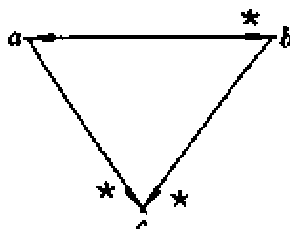


图 5-2

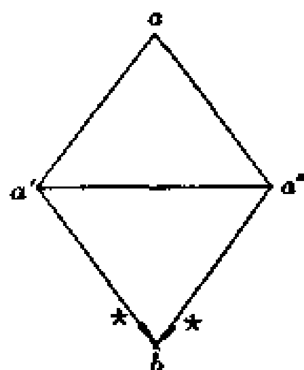


图 5-3

定理 1 如果 R 是终止的,则 R 是合流的,当且仅当 R 是局部合流的.

6. 完全性(completeness)

R 是完全的,如果对于任何两个不同的不可化简的词 a 和 b 都没有 $a\overset{*}{\leftrightarrow}b$.

如果 R 是完全的,则任何词都可化简到唯一的不可化简的词,因此任给两个词 α 和 β,能找到两个不可化简的词 α_0 和 β_0,使得 $\alpha\overset{*}{\leftrightarrow}\alpha_0$ 和 $\beta\overset{*}{\leftrightarrow}\beta_0$. 如果 $\alpha\overset{*}{\leftrightarrow}\beta$,则 $\alpha_0\overset{*}{\leftrightarrow}\beta_0$. 在 R 是完全的条件下,$\alpha_0\overset{*}{\leftrightarrow}\beta_0$ 当且仅当 $\alpha_0=\beta_0$. 因此词的问题在 R 是完全的条件下是可解的.

7. Knuth-Bendix 完备算法

设 $(\lambda_1,\sigma_1)\in R$, $(\lambda_2,\sigma_2)\in R$, u 是 λ_2 的非变量子项,记作 $\lambda_2[u]$, θ 是替换. 如果 $u\theta=\lambda_1\theta$, $\langle\lambda_2\theta[\sigma_1\theta],\sigma_2\theta\rangle$ 称为 (λ_1,σ_1) 和 (λ_2,σ_2) 的关键对;如果 $a\overset{*}{\rightarrow}b$,而且 b 是不可化简的,则记 b 为 a^*,如果 $(\lambda_2\theta[\sigma_1\theta])^*\neq(\sigma_2\theta)^*$,则此关键对叫发散的. 下面的算法用来确定给定的重写规则集是否是完备的.

Knuth-Bendix 算法:

步 1 在 R 中找出一个发散的关键对 $\langle s,t\rangle$,如果找不到就停止,并报告 R 是完备的.

步 2 如果 $R\cup\{(s^*,t^*)\}$ 是终止的,则 $R=R\cup\{(s^*,t^*)\}$;否则,如果 $R\cup\{(t^*,s^*)\}$ 是终止的,则 $R=R\cup\{(t^*,s^*)\}$;否则回答失败,并终止算法.

步 3 转步 1.

参 考 文 献

1 Chang C, Lee R C T. Symbolic logic and mechanical theorem proring. New York: Academic Press, 1973.

2 Knuth D E, Bendix P B. Simple word problem in universal algebras, in computational problems in abstract algebras. Pergamon Press, 1970, 263 ~ 297.

3 刘叙华,姜云飞. 定理机器证明. 北京:科学出版社,1987.

·计算机数学卷·

第 19 篇

并行与分布计算中的模型与算法

编　者　程　旭　刘建国　李晓明
审校者　袁崇义

目　录

引　言

在计算机科学与技术领域,并行处理(parallel processing)和分布处理(distributed processing),或者并行计算(parallel computing)和分布计算(distributed computing)是两个相关但又有明显区别的两个分支.

从应用的层次看,并行处理讨论的典型问题是:如果任务 A 在某个限定的时间 T 内由一台计算机做不完,能不能构造一个含有若干台计算机的系统 M,让每台机器做 A 的一部分,从而在时间 T 内使 A 得到完成.数值天气预报就是一个最好的应用例子,天气数据的采集总是在一天的某个时间(例如中午 12 点)才能得到,关于第二天的天气预报必须在晚上某个时间(例如下午 6 点)前作出,于是一定要在有限的时间里完成数据的处理和分析工作.并行处理技术在这里就可能发挥很好的作用:让 M 的每个计算节点处理所收集到天气数据的一部分,从而在总体上缩短计算时间,或者提高预报准确度(针对更为细致的数据),或者扩大预报的规模(例如从预报 24 小时内的天气扩大到预报 48 小时内的天气).这里所涉及的技术问题是多层次的,其最底层是如何将多个计算单元组织成一个有机的系统.如果这个系统只是为了解决一个(或一类)问题,我们可能建造一个十分有针对性的专用机.反之就要考虑能适应多种需要的通用系统.另一方面,人们还需要考虑如何用这样的系统来表达所要解决的问题,或者问题求解的步骤、如何分解该任务、各子任务之间的依赖关系该如何协调等等.这都属于这里的内容.

分布处理讨论的典型问题之一是:在一个由多台计算机构成的网络系统中,如何能花最少的时间使任务 $A_1,A_2,\cdots,A_n$ 都得到执行.这里人们追求的主要目标是系统的整体吞吐率,而不是某个具体任务执行的快慢.因此,任务的分派和调度、任务之间共享资源的管理等就是要考虑的技术问题.

将上述稍微换一个说法,可以看到并行处理和分布处理之间有如下松散的"对偶性".

并行处理:给定任务 A,求多机系统 M 和子任务 $A_1,A_2,\cdots,A_n$,使得 $A_1,A_2,\cdots,A_n$ 是 A 的分解,问 $A_1,A_2,\cdots,A_n$ 能否在时间 T 内在 M 上完成.

分布处理:给定多机系统 M 和任务 $A_1,A_2,\cdots,A_n$,求 A,将 A 看作是 $A_1,A_2,\cdots,A_n$ 的合成,问 $A_1,A_2,\cdots,A_n$ 能否在时间 T 内在 M 上完成.

离这种对偶性远一些,分布处理关心的另一类问题是在动态变化的系统中分布计算单元之间的信息交流.一种典型的情形是:在 Internet 上新接入一台计算机 A,就能够通过它向另一台早已存在于网上的计算机 B 发 E-mail.这里,B 和 A 不需要有什么直接的连接关系,B 也不需要被通知 A 的出现从而做什么特殊的准备.

一般地,讲并行处理和分布处理相关,是因为它们在应用中的表现形式通常都是多个协同工作进程的并发运行,这些进程运行于以某种处理能力在空间上分布于计算系统中.鉴于这样的计算机系统在体系结构上可分为共享存储和消息传递

两大类,共享存储和消息传递也就成了并行计算和分布计算模型分类的共同基础.再者,并行处理和分布处理常常共同存在于一个具体应用中,例如一个并行计算系统可能需要一个分布式文件系统的支持等.最后,为并行处理建立的计算模型和环境,也可能用于开发分布处理方面的应用,反之亦然(例如 MPI, Message Passing Interface是针对并行计算提出的,但也能用于开发 client/server 类应用系统,是典型的分布计算).

讲它们不同,首先是应用的目的不同.研究并行处理技术,追求的是利用和开发待求解问题的并行性,通过提供适当的含有多处理部件的计算环境,使得该问题求解的速度更快(和单处理部件的计算环境相比),在限定的时间内能解决的问题规模更大,在模拟物理世界的计算中获得更高的精度.分布处理的情形不同,它所要支持的是在一个系统中分布处理部件之间的信息交流和资源共享,其基本手段是提供处理部件之间协作的方式(体现为协议和分布式算法).从某种意义上讲,并行计算是面向问题的,分布计算是面向系统的.即并行计算是"我有了一个科学问题,看怎么能构造一个并行计算系统来解决它";分布计算是"现在有一个计算系统是功能分布的,看怎么能够使得系统的功能分布特性对用户透明(看不见)".

因此,研究并行计算和分布计算所关心的内容是很不同的.研究分布计算,不会关心诸如求解 FFT 的高效并行算法;而研究并行计算,也不会考虑在并行求解 FFT 时如果某个计算节点突然坏了该怎么办之类的问题.研究并行处理算法,常常是给定系统的拓扑结构,参与计算的进程数一定(或作为一个参数),考虑如何在系统进程之间划分计算任务和数据,以求得到时间和空间的高效率.研究分布处理算法,常常就要考虑进程数的不定,系统拓扑结构未知以及进程可能在运行中非正常终止;也考虑算法的复杂性,但其地位没有在并行计算中突出.在这个意义上,又可以说,并行处理更关心的是"好"与"坏";分布处理更关心的是"能"与"否".

如果认为 ILLIAC IV 是并行处理技术开始的象征,ARPA NET 的运行是分布处理技术开始的象征,那么并行与分布处理技术的发展已经有了 20 多年的历史.作为两种基本技术,并行处理和分布处理在实践中都得到了颇具成效的应用.并行处理技术在以数值计算为特征的应用中的典型例子是 FFT 算法的并行化,而在以数据处理为特征的应用中的典型例子是快速排序算法的并行化.至于分布处理技术,最有意义的应用应该是 Internet 了.IP 路由协议,就是其中最重要的分布式算法.

本篇分为三部分.第一部分为引言,介绍并行处理和分布处理的基本概念,比较它们的异同;第二部分介绍并行计算的模型和几个典型算法;第三部分讲分布式计算的模型和几个典型算法.20 多年来,人们在这两方面已积累了浩瀚的研究成果和丰富的文献资料,这里,所提供给读者的大概只能是在并行与分布计算领域人们在基础研究部分所关心的若干重要内容,关于并行和分布计算模型的不同风格的一种感受,以及相关算法设计与分析方法的典型思路.

1　并行计算的模型与算法

计算模型是设计和分析算法的基础.冯·诺伊曼(von Neumann)模型是一种串行计算机的通用计算模型.虽然不同串行计算机的处理器体系结构和存储组成之间可能存在差异,但它们都遵从相同的抽象计算模型.于是,针对冯·诺伊曼模型设计的程序就可以在任何一台串行计算机上相对有效地被编译和执行.特别是,在遵从冯·诺伊曼模型的不同计算机上,同一算法的渐进时间复杂性都是相同的.事实上,这一通用计算模型对串行计算机应用程序的开发和普及至关重要.如果对于每种新机器结构,都需要重新设计算法,则不仅软件研发的成本很高,而且软件本身也很难保证移植后的高效性.并行算法的速度不仅依赖于并行计算机的计算速度,而且与处理器之间的通信性能也有直接关系.目前,在并行计算机方面,还没发现一种普遍认同的通用计算模型,这一模型既可以在很大范围内概括并行计算机的基本特性,又能做到在许多实际机器上高效执行针对该模型设计的并行算法.因而本章首先简要介绍并行计算机的基本结构和模型,进而针对一种理想共享存储多处理器模型 PRAM 和两种网络模型展开对快速排序问题和 FFT 的并行算法的设计、分析.最后,对当前并行计算模型的发展进行简要讨论.

1.1　并行计算机结构和模型

1.1.1　计算机结构组织模型

首先,从并行计算机构造的角度来讨论结构组织模型.这些模型可以从控制机制、处理器粒度、地址空间组织和互联网络等方面加以区别.

1. 控制机制

传统上,人们主要根据指令和数据流的单一性或多重性对计算机系统结构进行分类.考虑如下笛卡儿积:

$$(\text{单指令流},\text{多指令流})\times(\text{单数据流},\text{多数据流}),$$

简记为(SI,MI) × (SD,MD),可得到四种基本结构:SISD、SIMD、MISD 和 MIMD.其中 SISD 就是冯·诺伊曼模型,MISD 尚没有对应的实际机器(注:有人认为"脉动阵列"属于该结构),现实中的并行计算机根据其处理单元是由单一集中控制器控制还是由独立的控制器控制分属于 SIMD 和 MIMD 两类,又其中 SIMD 专指在共同控制下以锁步方式工作的处理器阵列,它不包括向量流水线计算机.由于 SIMD 计算机在同一时钟周期所有处理单元只能执行相同操作,因而更适合于专用计算.适用于通用计算的并行计算机几乎都采用 MIMD 模型.

2. 处理器粒度

并行计算机既可以由大量处理能力较弱的处理器构成,也可以由较少的处理能力很强的处理器构成.并行计算机的**处理器粒度**定义为完成基本通信操作所需

时间与完成基本计算操作所需时间的比值.一般来说,处理器粒度较小的并行计算机适合处理通信频繁的算法;相反,处理器粒度较大的并行计算机适合处理通信不太频繁的算法.

3. 地址空间组织

多个处理器共同求解同一问题,处理器之间需要相互通信.消息传递和共享地址空间提供了两种不同的处理器相互通信模式.

在消息传递结构中,处理器采用消息传递互联网络连接,或每个处理器具有私用的本地存储器,或处理器之间的通信通过消息传递来实现,这种结构称为**分布式存储器结构**.采用消息传递的 MIMD 计算机通称为**多计算机**.

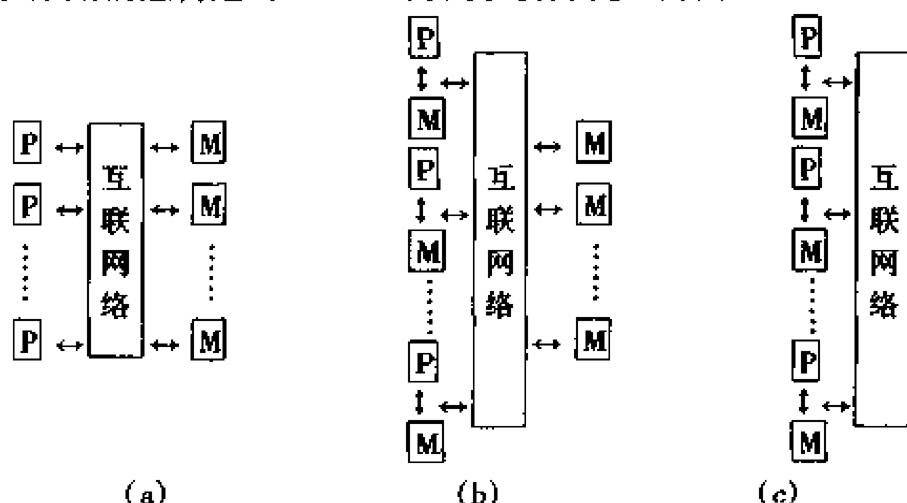


图 1-1

(a)UMA　(b) 具有本地和全局存储器的 NUMA　(c) 只有本地存储器的 NUMA

在共享地址空间结构(图 1-1) 中,处理器通过公共存储器的共享变量实现相互通信.采用这一结构的 MIMD 计算机通称为**多处理机**.根据处理器访问局部和全局存储器的时间,多处理机模型可分为:**UMA**(uniform-memory access) **模型**(图 1-1(a)) 和 **NUMA**(non-uniform-memory access) **模型**(图 1-1(b) 和图 1-1(c)).在 UMA 模型中,所有处理机对所有存储字具有相同的访问时间;在 NUMA 模型中,处理器对存储器的访问时间因存储字的位置不同而变化.在图 1-1(b) 中,本地存储器存放本地处理器所需的非共享数据,所有全局共享数据都存放在共享存储器中,处理器对本地存储器的访问时间小于对全局存储器的访问时间.图 1-1(c) 对图 1-1(b) 中的本地处理器进行概念扩展,某处理器对其它处理器所属存储器的访问可由地址映射机制直接映射到相应非本地处理器,这时共享存储器实际上分布在所有处理器的本地存储器上,所有本地处理器的集合组成了全局地址空间,可被所有的处理器访问,从而消除了物理上独立存在的共享存储器.

在具有 p 个处理器的共享地址空间多处理机上模拟消息传递多计算机,只需将全局地址空间划分成 p 个独立无重叠的分区,每个处理器分占一个分区;A 处理器向 B 处理器发送消息,可通过 A 处理器将所需传送的数据写入 B 处理器所属的存储器分区完成.这种模拟非常容易,而且效率也很高.在消息传递多计算机上也可模拟共享地址空间的多处理机,但需要发送和接收许多消息,因而代价很高.多

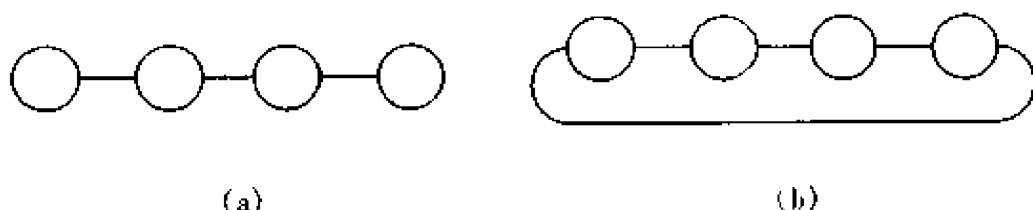
处理机在编程方面具有更高的灵活性.

4. 互联网络

多处理机和多计算机都需要选用不同的互联网络来将多个处理器和存储部件互联在一起.互联网络的拓扑结构决定网络特性.由于并行算法的速度不仅受制于算法蕴含的计算需求,而且受制于处理器之间的通信情况.因而并行算法的设计与互联网络通常直接关联.互联网络可分为静态和动态网络:**静态网络**由点对点通信链路而成,连接方式不可改变,也称为**直接网络**,它常用于消息传递型计算机中;动态网络采用交换开关和通信链路实现,通过开关交换通道可以动态改变结构,在不同结点之间建立动态链路,也称为**间接网络**,常用于共享存储空间计算机.典型的**动态网络**包括:交叉开关网络、基于总线的网络和多级互联网络.典型的静态互联网络包括:线性阵列网络、环和带弦环网络、循环移数网络、树形网络、星形网络、网格形(mesh)和环网形(torus)网络、超立方体(hypercube)网以及 k 元 n 立方体网络等.这里仅介绍线性阵列网络、网格形网络、超立方体网络和 k 元 n 立方体网络.

(1) 典型静态网络拓扑

● **线性阵列网络**　这是一个一维网络,其中 N 个结点用 $N-1$ 条链路连成一行(图 1-2(a));如果用一条附加链路将线性阵列的两个端结点连接起来,就形成一个环形网络(图 1-2(b)).



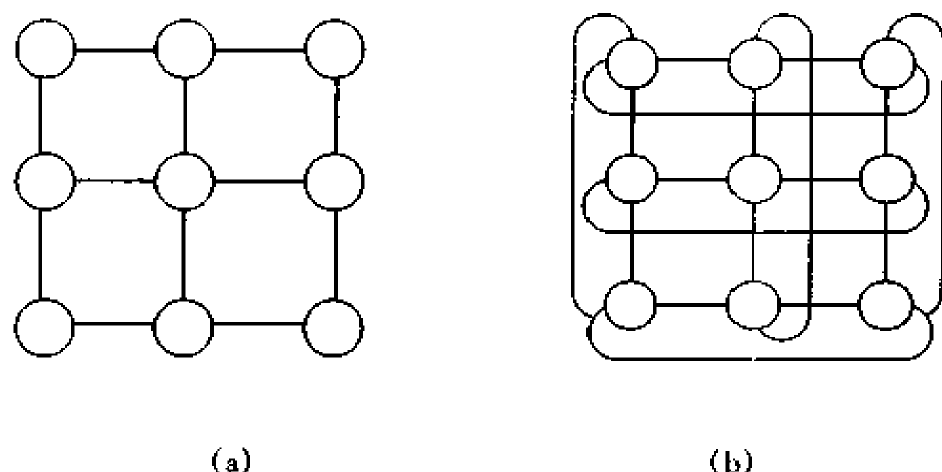
(a)　(b)

图 1-2

(a) 线性阵列网络　(b) 环形网络

● **网格形网络**　二维网格形网络是线性阵列网络的二维扩展,其中每个结点(除边缘结点)与它的四个相邻结点有直接链路(图 1-3(a)).如果二维网格形网络的对应边缘结点形成可回绕连接,即每行每列都有环形连接,则构成二维环网形网络(torus)(图 1-3(b)).同理,可以构造出 d 维的网格形网络和 d 维的环网形网络.

● **超立方体网络**　这是一种在每维上只有两个结点的多维网格形网络.一个 d 维的超立方体包括 $N=2^d$ 个结点.超立方体可以如下递归构造:0维超立方体只有单一结点;1维超立方体由互连两个0维超立方体产生;一个 $(d+1)$ 维超立方体通过互连两个 d 维超立方体的所有对应结点产生.可用二进制序列 $A=(a_{d-1}, a_{d-2},\cdots,a_2,a_1,a_0)$ 来标记一个 d 维超立方体的所有结点.标记方法为:1维超立方体的两个结点分别标记为(0)和(1);设一个 $(d+1)$ 维超立方体包含的两个 d 维超立方体 C 和 C' 中的结点都标记为 $(a_{d-1},a_{d-2},\cdots,a_2,a_1,a_0)$,则对该 $(d+1)$ 维超立方体中的属于 C 的结点标记为 $(0a_{d-1},a_{d-2},\cdots,a_2,a_1,a_0)$,而属于 C' 的结点标记为 $(1a_{d-1},a_{d-2},\cdots,a_2,a_1,a_0)$.图1-4示意了2维和3维超立方体及其结点的标记情

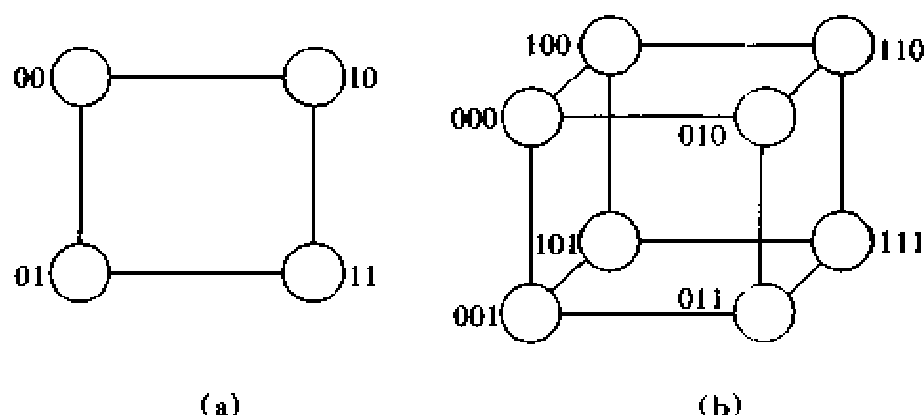


(a)　　　　(b)

图 1-3

(a)3 × 3 处理器的二维网格形网络　(b)3 × 3 处理器的二维环网形网络

况.



(a)　　　　(b)

图 1-4

(a) 二维超立方体网络　(b) 三维超立方体网络

超立方体网络具有如下重要性质:

(1) 两个结点之间存在直接链路,当且仅当这两个结点的标号仅有一位不同.

(2) 在一个 d 维超立方体中,每个结点与其它 d 个结点互连.

(3) 一个 d 维超立方体可以被划分成两个($d-1$) 维超立方体:任选结点标记中的某一位,该位为"0" 的所有结点及所属链路构成第一个分区,其它结点及所属链路构成第二个分区.由于 d 维超立方体的结点具有 d 位标号,因而共有 d 种不同的划分方法.

(4) d 维超立方体的结点的标号有 d 位,固定其中的任何 k 位,在其它($d-k$) 位具有不同标号的结点将构成一个包括 $2^{(d-k)}$ 结点的($d-k$) 维超立方体.

(5) 对于一个超立方体中的任意两个结点的标号分别为 s 和 t, s 和 t 的汉明距离(Hamming distance) 为 $s \oplus t$,则上述结点间的最短路径的通信链路数就是 s 和 t 的汉明距离.

- **k 元 n 立方体网络** d 维超立方体(也称为 2 元 d 立方体)是每维有两个结点的 d 维网格网络.环网是在一维上具有 k 个结点的网络.它们都是 k 元 n 立方体网络的极端特例.这里,d 称为网络的维数,k 称为基数(表示每维结点的数目).k 元 n 立方体的结点可用基数为 k 的 n 位地址 $A=(a_{n-1},a_{n-2},\cdots,a_2,a_1,a_0)$ 来表示,其中 a_i 表示第 i 维结点的位置.一个 k 元 n 立方体可以通过将 k 个 k 元 $(n-1)$ 立方体的所有具有相同标号的结点互连成环网的方式产生.

(2) 静态网络拓扑结构评价　首先,介绍几个常用的评价静态网络的成本和性能的参数.

- **结点度**　与结点相连接的边(链路)数目;结点度反映结点所需的 I/O 端口数,即反映了结点的成本.
- **网络直径**　网络中任意两个结点间最短路径的最大值.通常,路径长度用遍历的链路数来衡量.它从一个侧面反映网络的通信性能.
- **连通度(connectivity)**　网络的连通度是关于两个结点间存在路径数量的测度,弧连通度是评价网络连通性的一个测度,它定义为将该网络划分为独立两部分所需移去的最少边数.
- **等分宽度**　将网络等分成两部分时,需要移去的最少边(链路)数,它反映了网络对等的两个部分之间通信能力.
- **链路数**　网络的边(链路)的总数,它是评价网络成本的一个重要指标.

表 1-1 总结了具有 N 个结点的典型静态网络拓扑结构的特性.

表 1-1　具有 N 个结点的典型静态网络拓扑结构的特性

网络类型	最大结点度	网络直径	弧连通度	等分宽度	链路数
线性阵列	2	$N-1$	1	1	$N-1$
环网	2	$\lceil N/2\rceil$	2	2	N
星形网	$N-1$	2	1	1	$N-1$
全连接	$N-1$	1	$N-1$	$N^2/4$	$N(N-1)/2$
完全二叉树	3	$2\mathrm{lb}((N+1)/2)$	1	1	$N-1$
2 维网格网络	4	$2(\sqrt{N}-1)$	2	$\sqrt{N}$	$2(N-\sqrt{N})$
2 维环网形网络	4	$2\lfloor\sqrt{N}/2\rfloor$	4	$2\sqrt{N}$	$2N$
超立方体网络	$\mathrm{lb}N$	$\mathrm{lb}N$	$\mathrm{lb}N$	$N/2$	$N\mathrm{lb}N/2$
k 元 n- 立方体网络	$2n$	$n\lceil k/2\rceil$	$2n$	$2k^{n-1}$	nN

1.1.2　一种理想并行计算机模型——PRAM 模型

并行随机访问机器(PRAM:parallel random access machine)是针对共享存储计算机建立的零同步和存储访问开销的理想并行计算机模型.一台 PRAM 包括 P 个处理器和一个可被所有处理器进行均匀访问的无穷大的全局共享存储器.P 个处

理器同步地以读存储器、计算、写存储器循环方式运行. PRAM 对存储器的访问分为四种可选择的方式：

- **互斥读(ER)**　对于任意一个存储单元，在每个循环中最多只能有一个处理器对该存储单元进行读操作.
- **并发读(CR)**　在每个循环中，允许多个处理器从同一存储单元读取相同信息.
- **互斥写(EW)**　对于任意一个存储单元，在每个循环中最多只能有一个处理器对该存储单元进行写操作.
- **并发写(CW)**　在每个循环中，允许多个处理器同时对同一存储单元进行写操作，但必须制定策略解决写冲突问题.

根据上述四种可选方式，PRAM 可分为以下四种模型：

- **EREW-PRAM 模型**　禁止一个以上的处理器同时读、写同一存储单元. 这是限制最强的 PRAM 模型；
- **CREW-PRAM 模型**　允许并行对同一单元进行读操作，但必须串行化对同一单元的多个写操作；
- **ERCW-PRAM 模型**　允许并行对同一单元进行写操作，但必须串行化对同一单元的多个读操作；
- **CRCW-PRAM 模型**　允许并行对同一单元进行读操作和写操作，这是功能最强的 PRAM 模型.

并发读不会导致程序的任何语义问题，但并发写却可能导致写冲突，它可以采用以下策略之一来解决：

- **共用**　如果多个处理器对同一存储单元同时进行的写操作的内容都一致，才允许进行并发写；
- **任选**　从多个对同一存储单元同时进行的写操作的处理器中任选一个，允许该处理器完成写操作，并忽略其它的写操作；
- **优先**　将多个处理器事先排列出优先级，允许具有最高优先级的处理器完成写操作，并忽略其它的写操作.

基于 PRAM 模型，人们进行大量的并行算法设计和分析的理论工作，在本节后续的并行算法设计和分析中，主要利用 PRAM 模型来确定求解问题中所蕴含的并行性. PRAM 模型假设所有的处理器间的通信都是无代价的，忽视了对并行算法性能有重大影响的数据局部性特征，因而本节还将针对网格网络模型和超立方体网络模型介绍并行算法的设计和分析. 之所以选择网格网络结构，是因为：

(1) 网格结构可在相对较低的成本下扩展到很大规模，即可连接大量处理器.

(2) 许多应用算法可以自然地映射到网格网络.

(3) 虽然网格网络的网络直径较大，但随着消息的增加，对于采用直传(cut-through)和虫蚀(worm-hole)寻径的网络，网络时延并不明显；因而，在实际中，大量针对高连接度网络(例如超立方体)设计的并行算法在网格网络上也可取得较好的高性能.

(4) 许多商用并行计算机采用网格网络.

选用超立方体网络结构的原因是：

(1) 对一大类问题，最快的超立方体算法与最快的PRAM算法具有相同的渐进时间复杂性. 于是，超立方体网络不仅能体现这些应用的最大并行性，而且还可以利用其数据局部性.

(2) 对于许多问题，最好的超立方体算法也是其它网络(例如胖树、网格、多级网络等)的最好算法，从而使得针对超立方体设计的并行算法具有较好的适应性.

(3) 超立方体结构本身固有的递归性，使得它们适合设计很多并行算法.

1.2　性能和可扩展性的测度

串行算法的性能通常用执行时间来衡量，它可表示为关于输入规模的函数. 由于并行算法的性能不仅依赖于输入的规模，而且与并行计算机结构和处理器数量都直接相关，因而在评价其性能时就不能只考虑输入数据的规模. 并行计算系统通常是指并行算法和并行结构的组合. 下面简要介绍一些与评价并行算法有关的概念和测度.

- **并行运行时间**　指从并行计算开始到最后一个参与并行处理的处理器完成，执行之间的流逝时间，记为 T_p.

- **加速比**　指求解某一问题时，其最好的串行算法的串行运行时间与其被评价的并行算法在 p 个处理器上的并行运行时间的比值. 记为 S.

- **系统效率**　具有 p 个处理器的并行计算机系统的效率是并行加速比与处理器数目的比值. 记为 E，即 $E = \frac{S}{p}$.

- **代价**　指求解一个问题时，并行运行时间与使用的处理器数目的乘积.

如果求解某一问题的一种并行算法在并行计算机上的代价与求解该问题的已知最快串行算法的串行执行时间成线性比例，则称该并行算法(或该并行系统)代价最优. 代价最优的并行系统的系统效率记为 $\Theta(1)$.

可扩展的并行系统是指当处理器数目增加，处理问题规模也增大的情况下，可以保持系统效率恒定的系统. 可扩展性通常是并行计算机系统和并行算法设计所必须具备的特性. 因而研究随着处理器数目的增加，为保持系统的恒定，处理问题规模所必须保持的增长速率是非常有意义的.

- **问题规模**　指求解某一问题的最好串行算法在单处理器上运行时所需的基本计算操作的数量. 记为 W. 在后续讨论中，假设一个基本计算操作需要一个单

位时间.

● **并行系统开销**　指该系统的代价中包含的但已知最快串行算法在串行计算机上运行时所不需要的那部分代价.开销通常指并行处理中新增的同步和通信延迟所造成的无用计算,记为 T_0. T_0 是 W 和处理器数目 p 的函数,故也记为 $T_0(W, p)$.于是,

$$T_0 = pT_p - W.$$

由

$$T_p = \frac{W + T_0(W,p)}{p} \quad 及 \quad S = \frac{W}{T_p} = \frac{Wp}{W + T_0(W,p)},$$

可得

$$E = \frac{S}{p} = \frac{W}{W + T_0(W,p)} = \frac{1}{1 + T_0(W,p)/W},$$

则

$$W = \frac{E}{1-E}T_0(W,p).$$

令 $K = E/(1-E)$,可得

$$W = KT_0(W,p).$$

上式称为**恒定效率函数**,它反映了并行系统保持系统效率恒定的难易程度.

已知:当且仅当 $pT_p = \Theta(W)$,且 $pT_p = W + T_0(W,p)$ 时,一个并行系统代价最优,则

$$T_0(W,p) = O(W), \quad W = \Omega(T_0(W,p)).$$

上面两个公式表明,当且仅当一个并行系统的开销函数不能渐进超过问题规模时,该系统的代价最优.

1.3　快速排序的并行算法设计和分析

由于快速排序简洁、开销少,且平均复杂性最优,因而在串行计算机上广泛使用.

1.3.1　快速排序的基本思想和串行算法

快速排序算法采用分治的思想,将对一个大串排序的问题递归地分解为对两个较小的子串的排序问题.假设需要排序的 n 个元素分别存储在数组 $A[1\cdots n]$ 中.快速排序包括两部分:

(1) 分解. $A[q\cdots r]$ 序列被分解(重排)为两个非空的子串 $A[q\cdots s]$ 和 $A[(s+1)\cdots r]$,其中 $A[q\cdots s]$ 中的任意一个元素都小于或等于 $A[(s+1)\cdots r]$ 中的任意一个元素.

(2) 治理.对 $A[q\cdots s]$ 和 $A[(s+1)\cdots r]$ 递归运用快速排序处理.

这样,就可以保证整个序列被排序.

如何将 $A[q\cdots r]$ 序列分解(重排)为两个满足上述要求的非空的子串呢?通常的做法是从 $A[q\cdots r]$ 中选择一个元素 x,使用 x 来将 $A[q\cdots r]$ 分解为两个部分:包括所有小于或等于 x 元素的 $A[q\cdots s]$,与包括所有大于 x 的 $A[(s+1)\cdots r]$.其中,x

被称为主元(pivot).程序1描述了快速排序的串行算法.

程序1　快速排序的串行算法

```
1.   PROCEDURE QUICKSORT(A,q,r)
2.   BEGIN
3.     IF q < r THEN
4.     BEGIN
5.        x: = A[q];
6.        s: = q;
7.        FOR i: = q + 1 TO r DO
8.         IF A[i] ≤ x THEN
9.         BEGIN
10.          s: = s + 1;
11.          swap(A[s],A[i]);
12.         ENDIF
13.       swap(A[q],A[s]);
14.       QUICKSORT(A,q,s);
15.       QUICKSORT(A,s + 1,r);
16.     ENDIF
17.  END QUICKSORT
```

分解大小为 k 的序列的复杂性为 $\Theta(k)$.快速排序的性能受到分解方式的很大影响.在最坏情况,大小为 k 的串被分解为一个大小为1的子串和一个大小为 $(k-1)$ 的子串,此时的时间复杂性的递推公式为:$T(n)=T(n-1)+\Theta(k)$,可得 $T(n)=\Theta(n^2)$;在理想情况下,大小为 k 的串被分解为两个大小接近的子串,它们的大小分别为 $\lfloor k/2 \rfloor$ 和 $\lceil k/2 \rceil$,此时的时间复杂性的递推公式为:$T(n)=2T(n/2)+\Theta(n)$,可得 $T(n)=\Theta(n\mathrm{lb}n)$.第二种分解产生一种排序的最优算法.虽然快速排序的最坏时间复杂性为 $O(n^2)$,但是对于随机顺序的输入序列,快速排序所需的比较-交换操作的平均数为 $(1.4n\ \mathrm{lb}n)$,这是渐进最优算法.

1.3.2　快速排序的并行化

可以用多种方法对串行快速排序算法并行化.对于程序1所示程序的第14和15行,在每次调用QUICKSORT的过程中,数组已经被分解为独立的两个部分,每个部分都被独立地递归解决.于是,对分解后的两个子数组的排序是可以并行的.首先在单一处理器上运行QUICKSORT,当算法执行递归调用时(第14和15行),将两个子问题中的一个分派到另一处理器执行;这样每个处理器都继续使用快速排序算法对它的子数组进行处理,然后再将它派生的两个新子问题中一个分派到其它处理器处理.当子数组不能被分解时,该算法终止.在终止时,每个处理器拥有数组的一个元素,通过按一定顺序遍历所用处理器,就可以得知排序的结果(后面将讨论).这种并行算法使用 n 个处理器对 n 个元素进行排序.它的主要缺陷是使用单一处理器来完成对 $A[q\cdots r]$ 到 $A[q\cdots s]$ 和 $A[(s+1)\cdots r]$ 的分解.由于对 $A[1\cdots n]$

的分解只能由单处理器完成，因而该策略的运行时间下限受制于 $\Omega(n)$. 于是，它的处理器 - 时间乘积为 $\Omega(n^2)$，这不是代价最优的策略.

上述并行算法主要受限于对数组分解的串行化处理. 在后续策略中，可以发现数组分解的并行化是设计高效快速排序算法的关键. 分析最佳主元选择情况下的串行快速排序算法，其时间复杂性的递推公式为：$T(n) = 2T(n/2) + \Theta(n)$，其中，$\Theta(n)$ 的时间用于对数组进行分解. 将它与算法的全局时间复杂性 $\Theta(n \text{ lb } n)$ 相比，可以想像快速排序算法需要 $\Theta(\text{lb } n)$ 个执行步，每步需要 $\Theta(n)$ 完成数组分解. 于是，如果能够使用 $\Theta(n)$ 个处理器在 $\Theta(1)$ 时间完成数组分解，就有可能获得 $\Theta(\text{lb} n)$ 的全局时间复杂性，并且这是一个代价最优策略. 在不并行化数组分解工作的情况下，最好可以做到使用 $\Theta(\text{lb} n)$ 个处理器在 $\Theta(n)$ 时间完成对 n 个元素的排序.

上面的设想需要使用 $\Theta(n)$ 个处理器在 $\Theta(1)$ 时间将大小为 n 的数组分解为两个小数组，这对大多数并行计算模型是非常困难的. 目前，人们针对抽象 PRAM 模型发现了这类算法. 在网格形网络和超立方体网络连接的并行计算机上，由于通信开销的存在，事实上，分解步骤需要的时间总是大于 $\Theta(1)$.

1.3.3 CRCW PRAM 的并行快速排序算法

考虑在具有 n 个处理器的任选 CRCW PRAM 上完成对 n 个元素的并行快速排序.

执行快速排序可以被看为是构建一个二元树. 在该树中，主元是根结点，小于或等于主元的所有元素都在其左子树上，大于主元的所有元素都在其右子树上. 按序遍历该树就可获得排序的结果.

程序 2 CRCW PRAM 并行快速排序算法的二元树构造算法

```
1.  PROCEDURE BUILD_ TREE(A[1…n])
2.  BEGIN
3.    FOR each processor i THEN
4.    BEGIN
5.      root: = i;
6.        parent: = root;
7.      leftchild[i]: = rightchild[i]: = n + 1;
8.    END FOR
9.      REPEAT for each processor i ≠ root DO
10.    BEGIN
11.      IF (A[i] < A[parent_i]) OR (A[i] = A[parent_i] AND i < parent_i)
         THEN
12.       BEGIN
13.        leftchild[parent_i]: = i;
14.          IF i = leftchild[parent_i] THEN EXIT
15.        ELSE parent_i: = leftchild[parent_i];
```

```
16.        END IF
17.          ELSE
18.        BEGIN
19.          rightchild[parent_i]: = i;
20.            IF i = rightchild[parent_i] THEN EXIT
21.          ELSE parent_i: = rightchild [parent_i];
22.        END ELSE
23.      END REPEAT
24.    END BUILD_ TREE
```

程序2示意了针对 CRCW PRAM 模型设计的并行快速排序算法中的二元树构造策略，其中，需要排序的数组存放在 $A[1\cdots n]$ 中，数组元素 $A[i]$ 被分派给处理器 i. 数组 leftchild$[1\cdots n]$ 和 rightchild$[1\cdots n]$ 用于跟踪给定主元的后代. 对于每个处理器，本地变量 $parent_i$ 存放保存主元的处理器的标号. 开始，所有的处理器都希望将它各自的标号写入变量 root. 由于在任选 CRCW PRAM 模型中，并发写是任选的，因而只有唯一的一个处理器标号会被实际写入 root. $A[root]$ 的数值就被选为主元，root 被写入每个处理器的 $parent_i$；随后，所有拥有元素的数值比主元小或等于主元但又不是主元拥有者的处理器都希望将它各自的标号写入变量 leftchild$[parent_i]$，所有拥有元素数值比主元大的处理器都希望将它各自的标号写入变量 rightchild$[parent_i]$. 结果，只有两个数值分别被实际写入 leftchild$[parent_i]$ 和 rightchild$[parent_i]$，这两个数值就是在下一次迭代中拥有主元数据的处理器的标号，在下一次迭代中，这两个小数组被并行划分. 上述算法继续到选定 n 个主元. 当一个处理器所拥有的元素变成主元后，它就退出. 该算法每迭代一次，就在 $\Theta(1)$ 时间内产生新的一层树，因而该二元树构造算法的平均复杂性是 $\Theta(\mathrm{lb}n)$，与平均树高相同.

在生成二元树后，需要确定每个元素在排序后数组中的位置. 在 PRAM 模型下，使用 n 个处理器可在 $\Theta(\mathrm{lb}n)$ 时间完成对所生成的二元树的遍历，并确定每个元素在排序后数组中的位置，最后，再花费 $\Theta(1)$ 时间将每个元素放到正确的位置. 综上所述，上述算法在 n 个处理器的 PRAM 模型的平均运行时间为 $\Theta(\mathrm{lb}n)$，其处理器-时间乘积为 $\Theta(n\mathrm{lb}n)$，是一种代价最优策略.

1.3.4 超立方体模型的并行快速排序算法

超立方体有许多拓扑特性，例如，一个 d 维超立方体可以被分解为两个 $(d-1)$ 维超立方体，且分解后的每个子超立方体的每个结点都与另一子超立方体的一个相应结点互联. 于是围绕某一主元对一数组的排序，可以被看为对超立方体的分解，分解的结果是小于或等于该主元的所有元素在一个子超立方体中，大于该主元的所有元素在另一子超立方体中.

考虑在一台 $p=2^d$ 个处理器的 d 维超立方体互联的并行计算机上完成对 n 个元素的并行快速排序，每个处理器分配得到一个含 n/p 个元素的数据块，排序后数组元素顺序与处理器的标号顺序相同. 首先，选择一个主元，并将该主元广播到所

有处理器;每个处理器在接收到广播的主元后,将本地元素分解为两个子块,分别包含本地数据块中小于或等于该主元的所有元素和大于该主元的所有元素;然后,沿第 d 维的通信链路互联的处理器互换相应的数据块,使得处理器标号的第 d 位为0的 $(d-1)$ 维子超立方体包含原数组中所有小于或等于该主元的所有元素,处理器标号的第 d 位为1的 $(d-1)$ 维子超立方体包含原数组中所有大于该主元的所有元素;进而,每个子立方体递归地对所拥有的子数组进行排序.在进行 d 次(每维一次)上述分解后,原数据就按处理器结点标号序被排序.注意,上述工作并没有保证每个处理器上的数据块被排序.因而,每个处理器还需要利用快速排序串行算法完成对本地数据进行排序.程序3示意了基于超立方体的快速排序算法.

程序3　*d* 维超立方体并行快速排序算法(其中B是分配给每个处理器的 n/p 元素的子数组)

```
1.    PROCEDURE HYPERCUBE_ QUICKSORT(B,n)
2.    BEGIN
3.      id: = processor's label;
4.       FOR i: = 1 TO d DO
5.      BEGIN
6.         x: = pivot;
7.         partition B into B1 and B2 such that B1 ≤ x < B2;
8.         IF i^th bit is 0 THEN
9.         BEGIN
10.            send B2 to the processor along the ith communication link;
11.            C: = subsequence received along the ith communication link;
12.            B: = B1 ∪ C;
13.         ENDIF;
14.       ELSE
15.            send B1 to the processor along the ith communication link;
16.            C: = subsequence received along the ith communication link;
17.            B: = B2 ∪ C;
18.       ENDELSE
19.     ENDFOR
20.     sort B using sequential quicksort;
21.   END HYPERCUBE_ QUICKSORT
```

在上述超立方体快速排序算法中,主元的选择对算法的性能有很大影响.如果第一次选择的主元恰是数组中最大的元素,那么,在第一次分解后(按第 d 维),所有的元素都将集中到一个 $(d-1)$ 维子超立方体上,而另一个 $(d-1)$ 维子超立方体却没有任何元素.于是,在后续的排序工作中,最多只有一半处理器继续工作,而另一半处理器则空闲.这是一种极端情况,但却说明了上述算法存在的一个严重问题.理想情况是每次分解处理器都有大小为 n/p 的子数组.但是,如果主元选择不慎,不同处理器所拥有的子数组的大小就会出现很大差异,结果拥有较小子数组的

处理器将会空闲，以等待拥有较大子数组的处理器对其本地子数组完成划分. 这些子数组的大小差别越大，该并行算法的性能也就越差.

假设，在 d 次分解的每一次分解中，处理器 P_1 中存储的子数组的大小都增加 k 倍，其中 $1 \leqslant k \leqslant 2$. 于是，$P_1$ 处理器在上述 d 次分解中所花费的总时间为 $\Theta\left(\sum_{i=0}^{i=d-1} k^i n/p\right)$. 若 $k > 1$，总时间为 $\Theta((k^d - 1)n/p)$. 由于 $p = 2^d$，因而上式可简化为 $\Theta((p^{\mathrm{lb}k} - 1)n/p)$. 若 $k = 2$，则 P_1 分解所用的总时间为 $\Theta(n - n/p)$. 在进行 d 次分解后，P_1 上的子数组大小为 $2^d n/p = n$，P_1 需要进一步采用快速排序串行算法对 n 个元素进行排序. 在这种情况，使用 p 个处理器并不能改进性能，所有的排序事实上都在单处理器上完成. 若 $k = 1.1$，则分解所用的时间约为 $\Theta((p^{0.138} - 1)n/p)$，本地排序的子数组大小为 $n/p^{0.138}$. 理想情况，$k = 1$，则分解所用的时间为 $\Theta((n\mathrm{lb}p)/p)$，本地排序的子数组大小为 n/p. 由此可见，k 越大，算法的性能越差.

一种选择主元的方法是：在第 i 次分解，共分解出 2^{i-1} 个子立方体，其中每个子立方体中的一个处理器随机选择它的一个元素作为所属子立方体的主元. 这种方法类似于快速排序串行算法中的随机主元选取策略. 虽然这一策略可以在串行算法中使用，但是对于并行算法，它却不一定适合. 这是因为，在串行算法中，如果某次主元选取不当，分解出的两个子数组大小差别很大，但后续的主元选取都很合适，那么这次不当主元选择增加的工作量最多不会超过该主元划分的子数组的长度，这并不会对串行算法的性能产生太大损失. 但是对于并行算法，如果某次主元选取不当，则会在分解出的两个子立方体上导致负载的严重不平衡，这样即使后续主元的选择都很合适，也会严重影响后续步骤的性能.

如果开始时每个处理器上元素的分布是均匀的，那么就可以发现一种较好的主元选择策略. 在这种情况，开始时每个处理器上存放的 n/p 个元素恰好是所有 n 个元素的一个样板，即每个 n/p 元素的子数组的中值与所有 n 个元素的原数组的中值非常接近. 注意，在沿第 i 维进行分解时，2^{d-i} 个子立方体中的处理器独立交换数据！对于每个这样的子立方体，任意选择其中一个处理器，将该处理器拥有的子数组的中值作为该子立方体的主元. 这种分解不仅能够保持负载平衡(每个处理器大约有 n/p 个元素)，而且可以保持每个子立方体中元素的均匀分布. 因而，对后续子立方体可继续采用相同的主元选择策略.

在现实中，是否可以做到每个处理器的 n/p 个元素与整个数组元素具有相同的分布，需要视具体的应用而定. 对于一些应用，随机主元选择策略和中值主元选择策略都可以很好地工作，但对于另外一些应用，它们的性能却可能都不理想.

下面针对中值主元选择策略对并行算法进行分析. 该并行算法共执行 d 次迭代. 每次迭代执行三步：① 选择主元；② 广播主元；③ 分解数组.

在每次主元选择步，每个处理器找出其所属元素的中值. 如果每个处理器上的元素都已排序，这一工作可在 $\Theta(1)$ 时间完成. 为了避免处理器在每次查找中值时对其所属元素进行排序，可以对它的 n/p 个元素预排序并在每次迭代的过程中保持其排序. 为此，可在处理器间互换子数组时采用合并已排序子数组的方法. 另外，保持子数组处于已排序状态，还可以避免原算法最后所需的串行排序工作.

选择一个主元后，在第 i 次迭代广播主元需要 $\Theta(d-(i-1))$ 时间.因而，在所有 $d = \mathrm{lb}p$ 次迭代中，广播主元所需的时间为

$$\sum_{i=1}^{d} i = \frac{d(d+1)}{2} = \Theta(\mathrm{lb}^2 p).$$

算法的第三步需要三个阶段.第一阶段，每个处理器将所属数组分解为两个子数组.由于每个处理器拥有 $\Theta(n/p)$ 个元素，并且这些元素已排序，所以这一阶段需要 $\Theta(\mathrm{lb}(n/p))$ 时间.第二阶段，沿第 $(d-(i-1))$ 维通信链路互联的处理器互换相应的数据块.由于最多有 $\Theta(n/p)$ 个元素需要在处理器之间发送，因而该阶段需要 $\Theta(n/p)$ 时间.第三阶段，每个处理器将它拥有的子数组与新接收的子数组合并产生一个新的排序后的大子数组.由于两个被合并的子数组都已排序，所以该阶段需要 $\Theta(n/p)$ 时间.

每个处理器开始使用串行快速排序算法对它的 $\Theta(n/p)$ 个元素进行排序需要 $\Theta((n/p)\mathrm{lb}(n/p))$ 时间.于是，整个并行算法执行的时间为

$$T_p = \Theta\left[\frac{n}{p}\mathrm{lb}\frac{n}{p}\right] + \Theta\left[\frac{n}{p}\mathrm{lb}p\right] + \Theta(\mathrm{lb}^2 p).$$

由于排序 n 个元素的串行时间复杂性为 $\Theta(n\ \mathrm{lb}n)$，则加速比 (S) 和效率 (E) 为

$$S = \frac{\Theta(n\mathrm{lb}n)}{\Theta(n/p)\mathrm{lb}(n/p) + \Theta((n/p)\mathrm{lb}p) + \Theta(\mathrm{lb}^2 n)},$$

$$E = \frac{1}{1 + \Theta((\mathrm{lb}p)/(\mathrm{lb}n)) + \Theta((p\mathrm{lb}^2 p)/(n\mathrm{lb}n))}.$$

上式表明，当 $(p\mathrm{lb}^2 p)/(n\mathrm{lb}n) = O(1)$ 时，它是一个代价最优策略.于是，这个快速排序策略可以有效地适用于多达 $p = \Theta(n/\mathrm{lb}n)$ 个处理器的并行计算机.

另外，上式还表明由于主元广播而使恒定效率函数为 $\Theta(p\mathrm{lb}^2 p)$，这说明该快速排序算法具有很好的可扩展性.

1.3.5　网格模型的并行快速排序算法

下面针对二维网格模型，讨论其快速排序算法的设计.假设该网格中的处理器按行主次序进行编号，排序后的元素与这一编号次序相一致.设需要排序的数组的长度为 k，所有元素的数值都不相同，整个数组存放在编号为 $P_m, P_{m+1}, \cdots, P_{m+k}$ 的处理器上.快速排序的并行算法的基本思想是首先选择一个主元，利用它来分解（重排）数组，使得所有小于或等于主元的元素在网格的一部分，所有大于主元的元素在网格的另一部分，然后递归地对所得的子数组并行地实施上述重排策略.其中分解数据的工作包括以下四步：

(1) 随机地选择一个主元，并将其传送给处理器 P_m.处理器 P_m 利用网格的一个内嵌树将该主元广播给拥有数组元素的所有的 k 个处理器.

(2) 每个处理器都采集有关信息并通过上述内嵌树上传.具体地说，每个处理器都计算在它的左子树和右子树中小于主元的元素的数目和大于主元的元素的数目.由于每个处理器都知道主元的数值，因而就可以确定它所拥有的元素比主元小

还是大.每个处理器都将统计的这两个数值上传给它的父亲.由于网格的内嵌树是一个非完全树,一些结点可能没有左子树或右子树.在该步的最后,处理器 P_m 就知道了所有 k 个元素中小于和大于主元的具体元素数目.设比主元小的元素的数目为 s,那么在排序后该主元的处理器序号为 $(m+s)$;

(3) 将信息沿内嵌树下传,使得每个数组元素得知将被移动到小于主元部分和大于主元部分的具体位置.内嵌树的每个处理器都从它的父亲处接收到在小于主元部分和大于主元部分的下一个空位.根据每个处理器上存储的元素是小于还是大于主元,该处理器可将正确的信息进一步传送给它的子树.开始时,小于主元的位置是 P_m,大于主元的位置是 P_{m+s+1};

(4) 所有处理器根据接收的下传信息对所拥有元素进行置换,将每个元素移动到小于或大于主元的正确位置.

图 1-5 是在 4 × 4 网格上对 13 个元素的数组进行的分解:其中图(a) 是网格上处理器的行主编号:图(b) 是每个处理器中存储的元素数值;图(c) 是内嵌在网格一部分的树;图(d) 是在执行分解算法第二步后,在第一列上处理器的小于/大于主元的元素数目;图(e) 是在执行分解算法第三步后,在第一列上处理器下传的小于/大于主元的元素的目的处理器编号;图(f) 是在执行分解算法第三步后,所有元素的目的处理器编号;图(g) 是在一到一通信后,各处理器所拥有的元素.

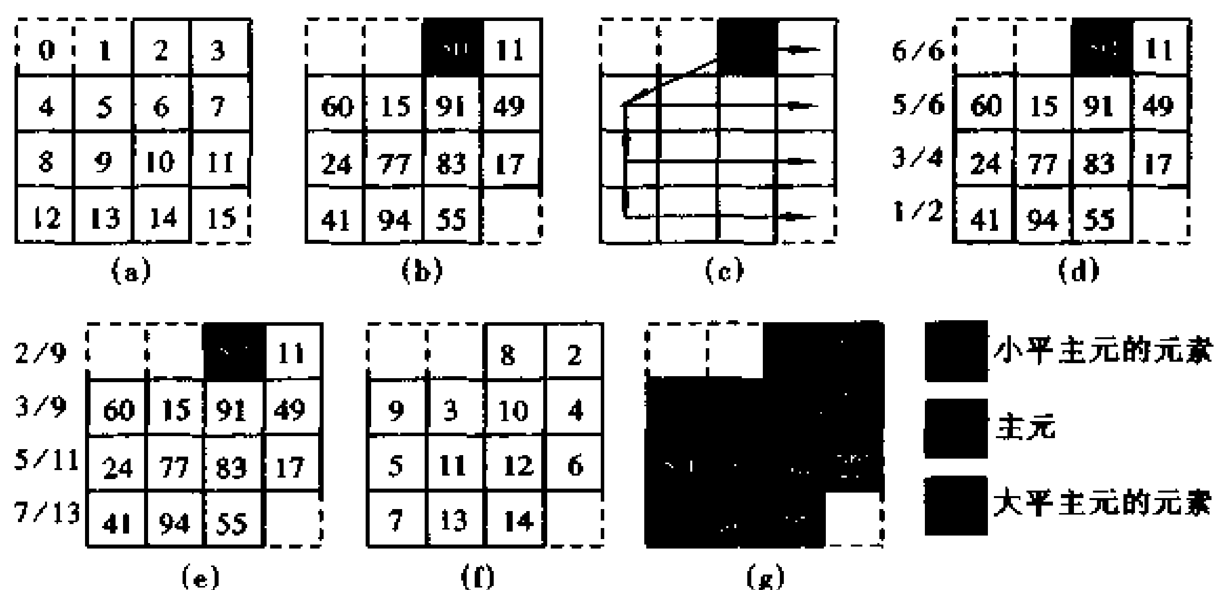


图 1-5

图 1-5 示意了上述分解算法的工作原理.并行快速排序算法的第一步是遍历网格中的内嵌树,由于从根结点到叶结点的最长路径为 $2\sqrt{n}$,该步需要 $O(\sqrt{n})$ 时间.在最后一步,在网格内对元素进行置换需要 $O(\sqrt{n})$ 时间.假设选择主元合适,该算法需要执行 $\Theta(\mathrm{lb}n)$ 次叠代.因而,该并行算法执行的总时间为 $O(\sqrt{n}\,\mathrm{lb}n)$.由于其处理器 - 时间乘积为 $O(n^{1.5}\mathrm{lb}n)$,因而这不是一种代价最优策略.可以通过减少排序所需的处理器数量,找到一种网格模型的代价最优快速排序算法.

1.3.6 并行快速排序算法的比较

表1-2 在p个处理器上对n个元素进行排序，不同并行算法的性能

并行计算模型	$E=\Theta(1)$时最大处理器数	并行执行时间	恒定效率函数
CRCW PRAM	n	$\Theta(\mathrm{lb}n)$	$\Theta(n\ \mathrm{lb}n\ n)$
超立方体			
最好情况	$\Theta(n/\mathrm{lb}n)$	$\Theta(\mathrm{lb}^2 n)$	$\Theta(p\,\mathrm{lb}^2 p)$
最坏情况	$\Theta(\mathrm{lb}n)$	$\Theta(n)$	$\Theta(p2^p)$
网格	$O(((\mathrm{lb}n)/(\mathrm{lb}\ \mathrm{lb}n))^2)$		$\Theta(p^{\sqrt{p}}\sqrt{p}\,\mathrm{lb}p)$

表1-2总结了三种并行计算模型下快速排序并行算法的性能. 其中，PRAM并行快速排序算法明显比超立方体和网格并行算法具有更好的可扩展性，它也是唯一的一种可在$\Theta(\mathrm{lb}\ n)$时间，在p个处理器上对n个元素进行排序的算法. 超立方体并行算法的性能与主元选择的质量有很大关系，如果主元选择最佳，该算法的可扩展性也就很好. 但是，如果主元选择不好，它就具有指数的恒定效率函数. 本节讨论的网格并行排序算法具有指数的恒定效率函数，因而在实际中只有当p较小时，才有效.

1.4 FFT的并行算法设计和分析

离散傅里叶变换(DFT)在时间序列分析、线性偏微分方程组的求解、数字信号处理、数字滤波等许多科学技术应用方面有着非常重要的作用.

1.4.1 串行FFT算法

n项序列$X=\{X[0],X[1],\cdots,X[n-1]\}$的DFT是一个$n$项序列$Y=\{Y[0],Y[1],\cdots,Y[n-1]\}$，其中

$$Y[j]=\sum_{k=0}^{n-1}X[k]\omega^{kj},\quad 0\leqslant j<n,$$

$$\omega=e^{2\pi i/n}.$$

根据以上公式计算每个$Y[j]$需要n次复数乘法操作，因而计算长度为n的整个Y序列的串行复杂性为$\Theta(n^2)$.

快速傅里叶变换(FFT)是一组快速计算有限傅里叶变换方法的总称. 它们可将DFT变换的串行复杂性降低到$\Theta(n\mathrm{lb}n)$.

假设n为2的幂，则

$$\begin{aligned}Y[j]&=\sum_{k=0}^{(n/2)-1}X[2k]\omega^{2kj}+\sum_{k=0}^{(n/2)-1}X[2k+1]\omega^{(2k+1)j}\\&=\sum_{k=0}^{(n/2)-1}X[2k]e^{2(2\pi i/n)kj}+\sum_{k=0}^{(n/2)-1}X[2k+1]\omega^{j}e^{2(2\pi i/n)kj}\end{aligned}$$

$$= \sum_{k=0}^{(n/2)-1} X[2k]e^{2\pi ikj/(n/2)} + \omega^j \sum_{k=0}^{(n/2)-1} X[2k+1]e^{2\pi ikj/(n/2)}.$$

令 $\tilde{\omega} = e^{2\pi i/(n/2)} = \omega^2$，则

$$Y[j] = \sum_{k=0}^{(n/2)-1} X[2k]\tilde{\omega}^{kj} + \omega^j \sum_{k=0}^{(n/2)-1} X[2k+1]\tilde{\omega}^{kj}.$$

上式说明，当 n 为 2 的幂时，可以将一个 n 项序列的 DCT 计算分解为两个 $(n/2)$ 项序列的 DCT 计算，并且可以递归地进行上述分解过程. 根据这一思想可以设计出程序 4 所示的 FFT 递归算法. 由于每次递归，输入序列都被分解为来两个对等的半部，因而该算法被称为基 2(radix-2) 算法.

程序 4　一维 FFT 递归算法

```
1.  PROCEDURE R_ FFT(X,Y,n,ω)
2.  IF (n = 1) THEN Y[0]: = X[0]ELSE
3.  BEGIN
4.     R_ FFT({X[0],X[2],…,X[n - 2]}, {Q[0],Q[1],…,Q[n/2]},n/2,
       ω²);
5.      R_ FFT({X[1],X[3],…,X[n - 1]}, {T[0],T[1],…,T[n/2]},n/2,
        ω²);
6.    FOR i: = 0 TO n - 1 DO
7.           Y[i]: = Q[i mod (n/2)] + ω^iT[i mod (n/2)];
8.  END R_ FFT
```

对于程序 4 所示的 FFT 算法，对于长度为 n 的输入序列，递归的最大深度为 $\mathrm{lb}n$. 在第 m 层递归，需要完成 2^m 次大小为 $(n/2^m)$ 的 FFT 计算. 但是，上述算法的串行计算复杂性为 $\Theta(n\mathrm{lb}n)$.

串行 FFT 算法也可以表示成迭代的形式. FFT 的迭代算法便于进一步讨论 FFT 算法的并行实现，程序 5 示意了与上述 FFT 递归算法对应的迭代算法. 该算法就是著名的计算一维 n 点基 2 FFT 的迭代 Cooley-Tukey 算法.

程序 5　一维基 2FFT 的 Cooley-Tukey 算法

```
1.  PROCEDURE ITERATIVE_ FFT(X,Y,n)
2.  BEGIN
3.     r: = lb n;
4.       FOR i: = 0 TO n - 1 DO R[i]: = X[i];
5.     FOR m: = 0 TO r - 1 DO /* Outer Loop */
6.       BEGIN
7.         FOR i: = 0 TO n - 1 DO S[i]: = R[i];
8.         FOR i: = 0 TO n - 1 DO /* Inner loop */
9.           BEGIN

/* Let (b0b1...br-1) be the binary representation of i */
```

```
10.            j: = (b0…bm-1 0 bm+1…br-1);
11.            k: = (b0…bm-1 1 bm+1…br-1);
12.            R[i]: = S[j] + S[k] × ω^(bm bm-1…b0 0…0);
13.         ENDFOR;/* Inner loop */
14.       ENDFOR;/* Outer loop */
15.       FOR i: = 0 TO n - 1 DO Y[i]: = R[i];
16. END ITERATIVE_FFT
```

上述算法有两个主要循环.完成 n 点 FFT,外层循环执行($\mathrm{lb}n$) 遍;对于外层循环的每次迭代,内层循环执行 n 遍.其中,内层循环执行的所有操作都是时间恒定的算术操作,因而该算法的串行复杂性为 $\Theta(n\,\mathrm{lb}n)$.在外层循环的每次迭代中,使用在上一次迭代中存储在序列 S 中元素来更新序列.对于首次迭代,用输入序列 X 的初始化序列 R.

程序 5 中的第 12 行是该 FFT 算法的关键步,用 $S[j]$ 和 $S[k]$ 来更新 $R[i]$.注意数组下标 j 和 k 的生成方法.假设 $n = 2^r$,由于 $0 \leqslant i < n$,故 1 的二进制表示需要 r 位.设下标 i 的二进制表示为($b_0b_1\cdots b_{r-1}$),则在外层循环的第 m 次迭代时($0 \leqslant m < r$),下标 j 等于将 i 的二进制表示的第 m 最大位(b_m)强置为“0”后的结果,下标 k 等于强置 b_m 为“1”后的结果.因而,若 $b_m = 0$,则 $j = i$;若 $b_m = 1$,则 $k = i$.而 j 和 k 的二进制表示仅有一位不同.

下面讨论如何在并行计算机上计算 FFT 的二元交换(binary-exchange)算法.在该并行算法中,只在处理器标号只有一位差异的处理器对之间发生数据交换.首先讨论如何在超立方体连接的并行计算机上实现二元交换算法.

1.4.2　超立方体模型的二元交换算法

首先假设并行计算机的每个处理器存储输入序列的一个元素,并产生输出序列的一个元素.

1. 每个处理器存储一个元素

利用 n 个处理器的超立方体来计算 n 点 FFT.图 1-6 示意了当 $n = 16$ 时,二元交换算法的通信模式(P_i 表示标号为 i 的处理器).

如图 1-6 所示,超立方体中处理器 $P_i(0 \leqslant i < n)$ 开始时存储 $X[i]$,结束时产生 $Y[i]$.在外层循环的 $\mathrm{lb}\,n$ 次迭代的每次迭代中,处理器 P_i 按照程序 5 的第 12 行更新 $R[i]$ 的数值.所有数组 R 的所有元素被并行更新.

注意,执行上述更新操作中,处理器 P_i 需要从标号与 i 仅有一位差异的处理器传送来 S 的一个元素,而根据超立方体的拓扑性质,所有处理器标号有一位差异的处理器对之间都存在一条直接链路,因而该并行 FFT 计算可以很自然地映射到超立方体模型上.

在该算法的 $\mathrm{lb}n$ 次迭代的每次迭代,每台处理器都执行一次复数乘法和加法,并与直接相连的一台处理器交换一个复数,每次迭代的计算量是恒定的,因而使用 n 个处理器的超立方体并行执行上述算法需要 $\Theta(\mathrm{lb}n)$ 时间.其处理器-时间乘积

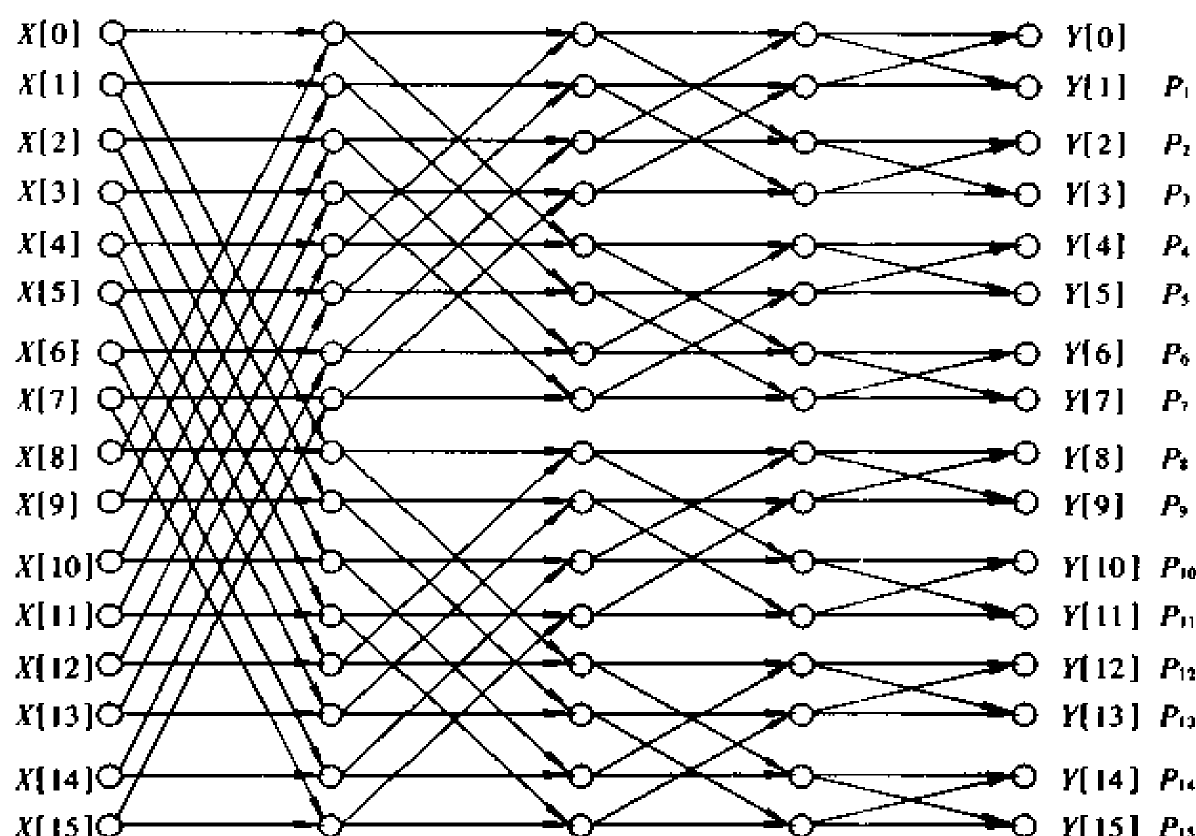


图 1-6

为 $\Theta(n\mathrm{lb}n)$,与最快串行 FFT 的复杂性相同,因而是一种代价最优策略.

2. 每个处理器存储多个元素

设在 p 个处理器的超立方体上计算 n 点 FFT,其中 $n > p$.假设 n 和 p 都是 2 的幂. n 项输入序列被划分成 p 个数据块,每块包含 n/p 个相邻元素,每个处理器分配到一个数据块.假设超立方体为 d 维(即 $p = 2^d$) 且 $n = 2^r$.

对于上述数据分配策略,对于任意 $i(0 \leqslant i < n)$,其二进制表示为 $(b_0 b_1 \cdots b_{r-1})$, $R[i]$ 和 $S[i]$ 都被映射到标号为 $(b_0 b_1 \cdots b_{d-1})$ 的处理器上,即任何数据元素的下标二进制表示的最高 d 位就是存放该元素的处理器的标号的二进制表示.在二元交换算法中,计算 n 点 FFT 外层循环需要进行 $r = \mathrm{lb}n$ 次迭代.在循环的第 m 次迭代,需要合并下标在第 m 最高位不同的元素,于是,在最早的 d 次迭代中合并的元素在不同的处理器上,而在后续$(r - d)$ 次迭代中合并的元素却在同一处理器上.

每次通信操作交换 n/p 个数据字.由于所有的通信都是在具有直接链路的处理器对之间发生,因而总的通信时间与路由选择的类型无关.假设通信延迟中的启动时间(startup time) 为 t_s,通信链路传送每个字的时间为 t_w,则整个算法在通信上所用的总时间为 $t_s\mathrm{lb}p + t_w(n/p)\mathrm{lb}p$.如果每次复数乘法和加法需要的时间为 t_c,则在 p 个处理器的超立方体上计算 n 点 FFT,二元交换算法的并行运行时间 T_p 为

$$T_p = t_c \frac{n}{p}\mathrm{lb}n + t_s\mathrm{lb}p + t_w \frac{n}{p}\mathrm{lb}p.$$

处理器-时间乘积为 $t_c n\mathrm{lb}n + t_s p\mathrm{lb}p + t_w n\mathrm{lb}p$,为了使该并行系统为代价最优,该乘积应为 $O(n\mathrm{lb}n)$;当 $p = O(n)$ 时,可满足上述要求.

该算法的加速比 S 和系统效率 E 为

$$S = \frac{t_c n\mathrm{lb}n}{T_p} = \frac{pn\mathrm{lb}n}{n\mathrm{lb}n + (t_s/t_c)p\mathrm{lb}p + (t_w/t_c)n\mathrm{lb}p},$$

$$E = \frac{1}{1 + (t_s p\mathrm{lb}p)/(t_c n\mathrm{lb}n) + (t_w \mathrm{lb}p)/(t_c \mathrm{lb}n)}.$$

n 点 FFT 的问题规模 W 为

$$W = n\mathrm{lb}n.$$

由于计算 n 点 FFT 最多使用 n 个处理器,则需要 $n \geqslant p$ 或 $n\,\mathrm{lb}n \geqslant p\mathrm{lb}p$ 来保证 p 个处理器处于忙碌.因而该并行 FFT 算法的恒定效率函数为 $\Omega(p\mathrm{lb}p)$.重写 E 的计算公式可为

$$\frac{t_s p\mathrm{lb}p}{t_c n\mathrm{lb}n} + \frac{t_w \mathrm{lb}p}{t_c \mathrm{lb}n} = \frac{1-E}{E}.$$

为了保持效率固定,则 $(t_s p\mathrm{lb}p)/(t_c n\mathrm{lb}n) + (t_w \mathrm{lb}p)/(t_c \mathrm{lb}n)$ 应该等于常量 $1/K$,其中 $K = E/(1-E)$.当 $t_w = 0$ 时,维持效率恒定的条件是:

$$\frac{t_s p\mathrm{lb}p}{t_c n\mathrm{lb}n} = \frac{1}{K},$$

$$n\mathrm{lb}n = K\frac{t_s}{t_c}p\mathrm{lb}p,$$

$$W = K\frac{t_s}{t_c}p\mathrm{lb}p.$$

上式是关于消息启动时间产生的通信开销的恒定效率函数.假设 $t_s = 0$,可以得到关于 t_w 产生的通信开销的恒定效率函数:

$$\frac{t_w \mathrm{lb}p}{t_c \mathrm{lb}n} = \frac{1}{K},$$

$$\mathrm{lb}n = K\frac{t_w}{t_c}\mathrm{lb}p,$$

$$n = p^{Kt_w/t_c},$$

$$n\mathrm{lb}n = K\frac{t_w}{t_c}p^{Kt_w/t_c}\mathrm{lb}p,$$

$$W = K\frac{t_w}{t_c}p^{Kt_w/t_c}\mathrm{lb}p.$$

综上所述,总的恒定效率函数为

$$W = \max\left\{p\mathrm{lb}p, K\frac{t_s}{t_c}p\mathrm{lb}p, K\frac{t_w}{t_c}p^{Kt_w/t_c}\mathrm{lb}p\right\}.$$

因而,该并行算法的可扩展性取决于超立方体的 CPU 计算速度与通信通道带宽的比值.在通信和计算速度平衡的超立方体上,该并行算法具有很好的可扩展性,并且当问题规模以 $\Theta(p\mathrm{lb}p)$ 的速率增长时,可以保持合理的系统效率.

1.4.3 网格模型的二元交换算法

假设在 p 个处理器的网格模型上计算 n 点 FFT,该网络有$\sqrt{p}$ 行和$\sqrt{p}$ 列,且$\sqrt{p}$ 为2的幂.令 $n = 2^r$ 和$p = 2^d$.假设处理器按行主元的方式编号,且数据的分布满足:下标为$(b_0b_1\cdots b_{r-1})$的元素映射到编号为$(b_0b_1\cdots b_{d-1})$的处理器上.

与上述超立方体模型的二元交换算法相同,在外层循环的 $\mathrm{lb}n$ 次迭代中,只有开始的 $\mathrm{lb}p$ 次迭代需要在编号差异一位的处理器对之间进行通信.需要注意的是,在网格模型中,进行通信的处理器对之间并不一定有直接链路.在 $\mathrm{lb}p$ 次需要通信的迭代中,每个处理器共需要与相应处理器进行 $\mathrm{lb}p$ 次通信,其中 $\mathrm{lb}\sqrt{p}$ 次通信在同行处理器间进行,$\mathrm{lb}\sqrt{p}$ 次通信在同列处理器间进行.同行或同列通信的处理器间的距离(穿越的链路数目)从1至$\sqrt{p}/2$.注意:对于给定的某次迭代,所有通信的处理器对之间的距离是相同的.于是,对于采用存储转发路由(注:指中间结点完全接收消息后,再向后续结点转发该消息)的网格模型,完成所有的同行通信的总时间为$\sum_{m=0}^{d/2-1}(t_s + t_w(n/p)2^m)$,完成所有的同列通信的总时间也为$\sum_{m=0}^{d/2-1}(t_s + t_w(n/p)2^m)$.假设每次复数乘法和加法需要的时间为 t_c,则在 p 个处理器的网格上计算 n 点 FFT,二元交换算法的并行运行时间 T_p 为

$$
\begin{aligned}
T_p &= t_c\frac{n}{p}\mathrm{lb}n + 2\sum_{m=0}^{d/2-1}\left(t_s + t_w\frac{n}{p}2^m\right)\\
&= t_c\frac{n}{p}\mathrm{lb}n + 2\left(t_s\mathrm{lb}\sqrt{p} + t_w\frac{n}{p}(\sqrt{p}-1)\right)\\
&\approx t_c\frac{n}{p}\mathrm{lb}n + t_s\mathrm{lb}p + 2t_w\frac{n}{\sqrt{p}}.
\end{aligned}
$$

该算法的加速比 S 和系统效率 E 为

$$
S = \frac{t_cn\mathrm{lb}n}{T_p} = \frac{pn\mathrm{lb}n}{n\mathrm{lb}n + (t_s/t_c)p\mathrm{lb}p + 2(t_w/t_c)n\sqrt{p}},
$$

$$
E = \frac{1}{1 + (t_sp\mathrm{lb}p)/(t_cn\mathrm{lb}n) + 2(t_w\sqrt{p})/(t_c\mathrm{lb}n)}.
$$

上述并行系统的处理器-时间乘积为$t_cn\mathrm{lb}n + t_sp\mathrm{lb}p + 2t_wn\sqrt{p}$.为了使该并行系统为代价最优,该乘积应为 $O(n\mathrm{lb}n)$;当$\sqrt{p} = O(\mathrm{lb}n)$,即 $p = O(\mathrm{lb}^2n)$ 时,可满足上述要求.关于消息启动时间产生的通信开销的恒定效率函数为 $\Theta(p\mathrm{lb}p)$.t_w 产生的通信开销的恒定效率函数为 $2K(t_w/t_c)2^{2K(t_w/t_c)\sqrt{p}}\sqrt{p}$,即当处理器数目增加时,问题规模必须以处理器数目的指数增长才能保持效率恒定.因而,该算法在网格模型上的可扩展性不理想.

1.5 评 述

PRAM 是最常用的描述和分析并行算法复杂性的理论模型.PRAM 在研究并行

计算的逻辑结构方面非常有用，实际中使用的并行算法大都以PRAM模型为研究起点，进而针对具体约束不断改进而成. 另外，可以在分布存储的机器上通过模拟PRAM支持PRAM并行算法的实现([7,8]). 文献[1,2,3,4]对PRAM并行算法进行了深入讨论. 虽然PRAM模型在研究应用问题内在并行性方面很有效，但它却忽视了许多现代并行计算机中非常重要的特点(例如，数据访问和通信代价通常决定中的程序执行时间)，因而许多优秀的PRAM并行算法在实际机器上的性能却很差. 以前，人们通常针对不同的网络拓扑结构来相对独立地研究通信问题，随着现代计算机中通信问题的重要性日趋明显，以及人们对通信事务中主要开销的认识越来越清晰，出现了一些用于分析通信代价的计算模型，例如BSP(bulk synchronous programming)模型([5])和log P模型([6])，这些模型试图替代PRAM而成为支持分布存储多处理并行算法设计和分析的实用模型.

并行计算BSP模型或粗同步并行计算机(BSPC: bulk-synchronous parallel computer)包括如下成分:

- 一些部件(component)，每个部件执行具体处理和/或存储功能;
- 一个路由器(router)，它在部件之间传递点对点消息;
- 以定期时间间隔对全部部件或一组部件提供栅栏同步(barrier synchronization)的设施，

如果将完成一次本地操作(对本地数据进行的基本操作)所需的时间定义为一个时间步，则可用下列四个参数来刻画BSP计算机的性能:

p:处理器数.

s:处理器速度，即每秒的时间步的数目.

l:同步周期(periodicity)，即在连续的同步操作之间的最少时间步的数目;

g:每秒所有处理器执行的本地操作的总数与每秒通信网络递送的数据字的总数的比值.

在BSP计算机中，计算由许多超步(superstep)的序列组成. 在每个超步，分配给每个部件一个任务，这个任务由一些本地计算步、消息传送和来自其它部件的消息(隐式)的组合构成. 在每过l时间单位后，进行一次全局检测以确定是否所有的部件都完成了该超步. 如果已完成，机器就开始处理下一超步;否则，下一个l时间单位就分配给该尚未完成的超步.

根据下列方法可确定BSP算法中一个超步S的复杂性: 令L为S期间任何进程执行的本地计算步的最大值，h_1为S期间任何处理器发送消息的最大值，h_2为S期间任何处理器接收消息的最大值. 则S的代价为$\max\{l, L, gh_1, gh_2\}$时间步.

LogP模型是一种每个处理器都通过点对点消息通信的分布存储多处理器模型，与BSP模型一样，该模型详细说明互联网络的性能特性. 但并不描述网络的结构. 该模型的主要参数如下:

L:在源模块和目标模块之间传递一个包含一个存储字信息(或少量存储字)的消息的时延(latency)上限.

o:开销(overhead)，定义为对于每次消息传递，处理器介入传送或接收该消息

的时间长度;在这段时间内,处理器不能进行其它操作.

g:间隔(gap),定义为在一个处理器上,连续消息发送或消息接收之间的最短时间间隔. g 的倒数对应于可用的每个处理器的通信带宽.

p:处理器/存储器模块的数目.

假设本地操作需要一个单位时间,称为一个周期(cycle);另外,假设网络的容量有限,即任何时间、从任何处理器到任何处理器最多只能传送$\lceil L/g \rceil$的消息.如果处理器试图传送的消息超过了这一限制,就必须暂停等待.

参数 L、o 和 g 用处理器周期的倍数来计量.该模型为异步模型,即处理器之间异步工作,任何消息的具体时延不可预测,但在没有暂停等待时,其时延上限为 L. 由于时延可变,因而消息的抵达次序可与发送次序不同.基本模型假设所有的消息都很小.在分析算法中,关键的度量指标是任何处理器使用的最长时间和最大空间.

BSP 模型和 LogP 模型都试图揭示与通信事件相关联的最重要的代价,例如时延、带宽和开销,从而使并行算法设计人员可以对上述参数进行调整来开展并行算法的比较性分析.BSP 模型还是一个用于探讨通信和并行性能非常好的构架.

另外,过去人们非常强调通信代价的建模,现在则更重视在结点处(通信消息的端点)的代价,因而在这些结点上消息的数量和竞争变得比到网络拓扑的映射更加重要.事实上,BSP 模型和 LogP 模型都完全忽视具体的网络拓扑结构,而在建模中将网络延迟设为常量!

BSP 和 LogP 模型在为设计和分析并行算法提供更加真实的计算模型上迈出了很重要的一步.通过改变这些模型中关键参数的数值,人们就能确定并行算法在一定范围的机器结构上的性能到底怎样,但是,由于并行计算系统是并行结构和并行算法的组合,人们可以用一组参数来对并行结构建模,但目前应该用哪些参数来对并行算法和应用问题建模还是有待进一步研究的问题.

2 分布计算的模型与算法

本节的内容主要来源于参考文献[9]和[11].一个分布计算系统,例如一个计算机网络,通常由若干具有一定处理能力的计算机(称为节点)按照某种拓扑结构连接而成.这个结构动态可变,其节点的增加和撤消(失效)事先不可预知.一个节点因此也就不一定知道整个系统的结构,甚至不知道参与算法执行的进程的个数,它通常只知道与它直接相连的节点.算法的输入可能从多个不同的节点得到,节点之间的通信发生的时机也是不确定的.这就是分布计算的典型物理环境.分布计算所要研究的,就是如何有效地支持这样一个系统中节点之间的信息交换和整个系统的资源(计算、存储、信息)共享."不确定性"是分布计算系统有别于并行计算系统的主要特征.

宏观上看,有三个方面的问题需要考虑.

一是系统的模型,即上述系统的有效的抽象表示.所谓有效,是要求模型能支

持分布式算法的设计与分析．在针对冯·诺依曼体系结构的算法设计与分析领域，由于有图灵可计算性理论作为坚实基础，我们有 RAM 模型等，它们被实践证明是有效的．相比之下，针对分布处理系统，人们提出了多种模型，但尚未得到如同 RAM 那样普适的东西．

二是在模型上算法分析的方法，即如何说明一个分布式算法的正确性和它实际应用于分布系统的有效性．和并行算法比起来，作为其它分布式应用系统基础支持的分布式算法看起来通常都是很简短的，但由于其所要面对的多种不确定性，分布式算法的行为往往不容易把握．即使是相同的输入，由于并发自主进程的参与，算法也可能表现出多种不同的行为，要得知其性质往往相当复杂．因此，要精确地给出分布式算法的行为通常是办不到的．于是，人们往往能做的是理解其行为的某些性质．我们还将看到，如果说设计并行算法关心的是加速比、处理器利用率等性质，对分布式算法人们关心的是不同的东西，例如算法的运行是否会造成破坏、每个参与算法运行的进程是否能得到公平的机会向前推进等．这就是所谓安全(safety) 和活跃(liveness) 性质．

安全和活跃实际上是两类性质的总称，不同的情形含义有所不同．在串行系统中，安全性主要体现在所谓部分正确性，即给定正确的输入、希望得到正确的输出(不一定关心系统对错误输入的行为)；活跃性主要体现在终止性，即程序最后应该结束．在并发计算系统中，情形要丰富得多．互斥、有限的消息延迟等都可看作是安全性的要求；没有"饥饿" 进程可以看成是活跃性要求．

所谓方法，在这里指的是证明分布式计算系统(算法) 性质的一般性手段和思路．例如，证明同步系统具有某种性质最重要的方法是证明不变断言(invariant assertion)．不变断言是系统状态在每次执行中，每轮执行后都成立的某个性质．证明时常对轮数 r(从 0 开始) 运用归纳法．另一个重要的方法是证明模拟关系(simulation relation)：如果算法 A 和另一个算法 B 以同样的输入开始，有同样的失败模式，每轮之后有相同的状态，则算法 B 具有的性质算法 A 也同样具有．这样便间接地用 B 证明了 A．

三是分布式算法设计本身．在分布式系统中，有一些基础性问题，例如在系统的节点中挑选出满足某种性质(例如编号最大) 的节点等．针对这些问题的分布式算法，不仅有基础性应用，而且也是对上述模型和方法的检验．

2.1　模　　型

一般来讲，分布式计算模型通称为**进程模型**(process model)，其中计算活动发生在若干并发执行的顺序进程之中．在这个顶层概念下，不同的进程模型由它们的进程间通信机制来区分．有两种基本的通信机制：一是消息传递(message passing)，二是共享变量(shared variable)．所对应的分布式系统分别称为网络系统和共享存储系统．

无论是消息传递还是共享变量，进程间的同步(synchronization) 问题是它们的一个共同基本概念，因为它是多个自主的进程形成一个有机系统的基础．"同步"

一词在不同的场合含义有别，有时，它指的是系统的组成部分以“齐步走”(lock-steps)或“步进”的方式共同向前推进(例如SIMD计算机系统的运行方式)，而这里的同步指的是系统的各组成部分(进程)按照一定的次序触发其事件的要求.要满足计算所提出的同步要求，有两个不同方向的情况可能需要考虑：一是竞争(contention)，二是协作(cooperation).竞争的典型例子发生在对共享变量访问上，不做特别的处理，系统的状态可能就是不确定的.以临界区为代表的互斥(mutual exclusion)技术就是处理竞争的经典手段.协作的经典例子是所谓有限缓冲(bounded buffer)问题.一个有限定长的缓冲区供一个生产者和一个消费者使用，生产者向缓冲区中放产品，但若缓冲区满了则要等待；消费者从缓冲区中取出产品，但若缓冲区空了则要等待.竞争与协作的本质区别在于竞争中的一个进程可以不依赖其它进程而向前推进，而协作中若一个进程停止运行了，另外的进程迟早也会被迫停止.

竞争与协作问题在分布式系统中几乎无处不在.在消息传递模型中，有一个所谓全局一致性(global consistency)问题，例如在一个分布式银行系统中，人们希望无论在哪里发生账目往来，所得到的状态(比方余额数)都是相同的.全局一致性问题就是要保证所有进程都有一个关于分布式系统中的当前全局状态的一致认识.

将分布系统的运行看成是事件通过状态相互作用的过程.要定义一个全局状态，必须有所有事件的一个全序.在上述银行例子中，取款是一事件，这一事件能否发生取决于余额，还取决于当前是否有存款(又一事件)正在发生.在异步消息传递模型中，不存在一个自然的事件全序，只有偏序，即如果 a 和 b 之间有信息流使得 a 影响 b，则事件 a 被认为在事件 b 之前.要定义一个全局状态，就要将这种偏序嵌入到某个全序中.这样一来，全局一致性问题就可以被化简为保证所有进程都取相同的事件全序的问题，从而就有了一个关于系统全局状态的一个公共定义.获得这种公共全序的方法之一是通过所谓逻辑时钟(logical clock).它是每个进程维护的一个计数器，如果事件 a 应该发生在 b 之前，则 a 的逻辑时间就在 b 的前面.逻辑时钟由时标(timestamp)来实现，它是发送者附着在每个消息上的逻辑时钟值.

除了消息传递和变量共享的区别外，对分布式计算模型的另一种主要划分是同步(synchronous)和异步(asynchronous).正如前面提到的，“同步”一词是多义的.这里，“同步”意味着“齐步走”，系统中的进程步调一致地进行计算和通信.而“异步”并不是讲进程之间没有关系，只是强调进程之间的通信在时间上事先不可知.下面给出几个常见的模型.

2.1.1 同步网络系统模型

这是关于分布式系统最简单的一个模型.系统被看成是一个进程的网络，两个进程在网络中相邻，表示它们的通信可以直接发生.在操作方式上，所有进程都按一种统一的步进方式向前推进.一个同步网络系统在结构上可以形式化地表示为一个有向图 $G=(V,E)$，节点集 V 由被称为进程的计算元素组成，边集 E 连接这些进程.为讨论方便，引入以下表示符号：

n：$n = |V|$，有向图 G 中的节点总数；

out-nbrs$_i$：进程 i 的出向邻居，即从 i 到 out-nbrs$_i$ 的每一个节点均有一条有向边；

in-nbrs$_i$：进程 i 的入向邻居，即从 in-nbrs$_i$ 的每一个节点到 i 均有一条有向边；

nbrs$_i$：当 G 为无向图时，nbrs$_i$ 用来表示进程 i 的邻居；

distance(i,j)：从进程 i 到进程 j 的最短有向路径的长度．如果不存在这样一条路径，则 distance$(i,j) = \infty$；

diam：图 G 的直径，即最大的 distance(i,j) 值（对图中所有的 (i,j) 对而言）；

M：进程之间交换的消息所组成的集合，null 表示空消息．

对 G 中每个节点 $i \in V$，对应的是含有如下成分的进程：

states$_i$：状态集合，不一定是有穷集合；

start$_i$：起始状态或初始状态，是 states$_i$ 的非空子集；

msg$_i$：**消息产生函数**，将 states$_i$ × out-nbrs$_i$ 映射到 $M \cup \{\text{null}\}$．其作用是为进程 i 的每一个状态产生一个节点 i 发给每个出向邻居的消息（如果有的话）；

trans$_i$：**状态转移函数**，将 states$_i$ 和由 $M \cup \{\text{null}\}$ 的元素组成的向量（以 in-nbrs$_i$ 为索引）映射到 states$_i$．其作用是说明每个状态在收到从入向邻居来的消息时将转移到什么新状态．

和 G 中的每条边相对应的是一条通道（channel），或称为链路（link），它可看作是一个用于存放 M 中消息的场所．通道中任何时候最多只能存放一条消息．

同步网络系统开始执行时，所有进程处于任意初始状态，所有通道为空．开始执行后，所有进程以步进的方式重复地执行下述两个步骤，直到进程到达停止状态（例如，唯一的状态转移是自环，并且不能产生任何新消息的状态）：

步 1　将消息产生函数应用在当前状态上，产生将要送往所有出向邻居的消息，并将这些消息置入相应的通道中．

步 2　将状态转移函数应用在当前状态和接收到的消息上，获得新状态．将所有的消息从相应的通道中取出．

上述两个步骤合起来称做模型执行的一轮（round）．注意，对 msg 和 trans 的复杂性没做任何限制，不同的进程当然可能有不同的消息产生函数和不同的状态转移函数．所谓"同步"体现在所有的进程一起按上述的轮次向前推进．

为刻画进程的 I/O，可将进程的起始状态看成是由某些变量的集合所决定的（变量取值的不同组合对应不同的状态），这个集合中含有一些和输入对应的变量．由于进程可能有多个输入，在定义中允许多个起始状态是有意义的．也就是说，这样就可以用进程的起始状态来表达输入．输出用指定的输出变量表示，输出变量只记录第一次对它进行写操作的值（即它是一次写变量），但是可以读多次．

系统的状态赋值（state assignment）是指为给系统中的每个进程赋予一个状态值，消息赋值（message assignment）是指为每个通道赋予一个消息（包括 null）．系统的一个执行（execution）定义为一个无穷序列：

$$C_0, M_1, N_1, C_1, M_2, N_2, C_2, \cdots,$$

其中 C_r 是状态赋值，代表 r 轮后系统的状态，C_r 常常被称做时间 r 时的状态赋值．M_r、N_r 是消息赋值，分别代表 r 轮时系统发送和接收的消息．如果进程 i 在两个系统执行 α 和 α' 中有相同的状态序列，相同的出向消息序列和入向消息序列，则称 α 和 α' 对进程 i 而言不可区分(indistinguishable)．

进一步地，如果进程 i 在 α 和 α' 中直到 r 轮都有相同的状态序列、相同的出向消息序列和入向消息序列，则称 α 和 α' 对进程 i 而言经过 r 轮无区别．无区别的定义可以扩展到两个不同同步系统的比较．

有时，不同的进程执行开始的时间不同．为了表示同步系统中进程可能在不同的轮次中开始执行，可在网络图中引入一个特殊的节点——环境节点(environment Node)．环境进程的任务是向所有其它节点发送特殊的唤醒消息(wakeup)．每个其它节点的开始状态为休眠状态，即它只能在收到环境进程发来的唤醒消息或者从其它进程收到非空消息的情况下，才可以转移到其它不同状态．

在此模型中，考虑两种类型的失败：进程失败(process failure)和链路失败(link failure)．进程失败又分两种情况：停止失败(stopping failure)——进程在执行的过程中失败，这可以在执行步骤 1 或步骤 2 的某些实例之前或之后，或者在执行步骤 1 的过程中(这时只有一部分消息被置入通道中)；拜占庭失败(Byzantine failure)——进程可产生任何消息及可进行任何状态转移，而不一定遵守消息产生函数和状态转移函数规定的规则．链路失败由链路丢失消息引起．进程试图在步骤 1 中向通道中放置消息，但是出错的链路并不记录消息．

上述定义的同步网络模型是确定性的(deterministic)，因为消息产生函数和状态转移函数都是单值函数．如果给定一组起始状态，计算便会以唯一的方式展开．但是，有时允许进程以某种概率分布进行随机选择是有用的．因为基本的同步系统模型不允许随机性，在消息产生函数和状态转移函数之外增加一个新的随机函数，以代表随机选择．对于每一个节点 i，增加一个元素 $rand_i$，对于每个状态 s，$rand_i(s)$ 是在 $states_i$ 子集上的概率分布．在执行的每一轮，随机函数 $rand_i$ 首先用来选择新状态，然后 msg_i 和 $trans_i$ 函数再像通常那样在新状态上应用．这样，系统的一个执行便表示为这样一个无穷序列：

$$C_0, D_1, M_1, N_1, C_1, D_2, M_2, N_2, C_2, \cdots, D_r, M_r, N_r, C_r, \cdots$$

其中 C_r 和 D_r 是状态赋值，M_r 和 N_r 是消息赋值，D_r 表示 r 轮随机选择后新的进程状态．随机系统的计算结果往往是用概率分布来表示的，这种概率性结果应该在所有输入、所有系统失败模式下都成立．为了表示输入和失败模式，通常使用一个虚拟的实体——敌手(adversary)来控制输入的选择和失败的发生．概率性的结果保证系统在任何允许的敌手存在的情况下都能够正常运行．

对于一个符合同步网络模型的同步分布式算法，常常考虑两种算法复杂度：**时间复杂度**和**通信复杂度**．时间复杂度用得到所有输出或使所有进程停止需要的轮数来度量．如果系统允许可变的起始时间，时间复杂度从唤醒消息第一次出现时算起．通信复杂度用发送的所有非空消息数来度量．有时也考虑消息中的位数．对所有分布式算法(不仅是同步网络)而言，时间复杂性在实践中比通信复杂性更重要．当网络通信量足够大，造成了系统阻塞，降低了处理速度时，通信复杂性就显得

重要起来. 但另一方面,网络带宽是由多个算法共享的,如果一个算法产生的消息造成了网络的拥塞,其它算法的执行也都会受到影响. 要量化这种相互的影响是很困难的,于是人们通常做简单处理:尽量减小算法本身的通信复杂度.

2.1.2 I/O 自动机模型

从这里开始,下面两个模型是有关异步系统的. 如同同步模型,异步模型的描述本身通常也是不困难的. 费解之处主要发生在讨论所谓活跃(liveness)条件时,即考虑如何能保证系统的各个部分都能不断得到运行的机会. 然而,和同步模型相比,由于在事件次序上的不确定性,异步模型编程较困难. 不过,由于异步模型对时序的假设要低于典型分布式系统实际能保证的,针对异步模型设计的算法通常具有一般性和可移植性 —— 它们在有任何时序行为的网络中都能正确运行.

输入/输出自动机(input/output automaton),后面简称 I/O 自动机,是一个针对异步计算的形式化模型. 它是构成其它异步模型的一种基础.

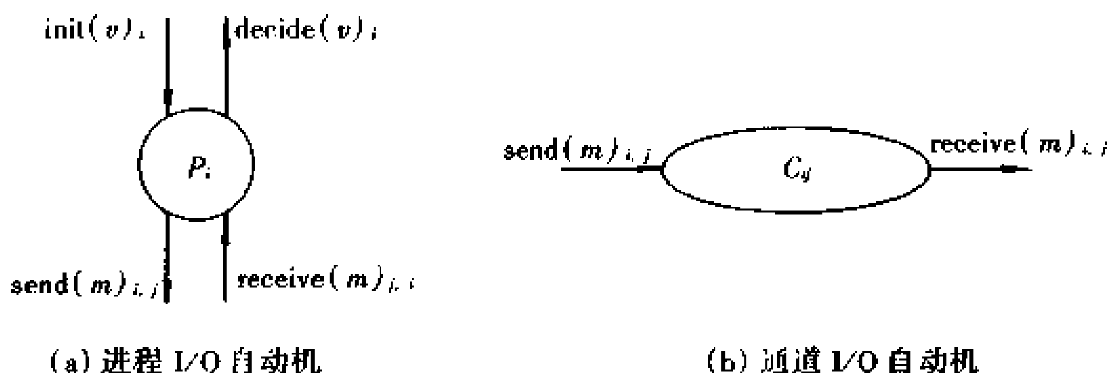
这个模型具有很强的通用性,几乎可以用来描述任何异步并发系统,包括后面将要介绍的异步共享存储系统. 就它自己而言,I/O 自动机模型不含什么结构信息,这使得它可以被用来作为许多不同类型分布式系统模型的基础. 这个模型提供了一种精确的方式来描述和推理系统中相互作用并以任意相对速度运行的部件(例如,进程或通信通道)的行为.

从I/O 自动机的定义和它的执行机制开始,然后定义一种合成操作(composition operation). 通过合成,多个I/O自动机能够组合形成一个更大的自动机来表示一个并发系统. 以下将展示这种合成操作所具有的一些漂亮性质,然后介绍公平性(fairness)的概念,它追求在一个系统中的所有部件应该得到"公平的"机会来向前推进。

I/O 自动机可以作为分布式系统部件的一种模型,这些部件能和系统中其它的部件相互作用;同时它也能作为整个系统的模型(如后面将看到的异步共享存储系统). I/O 自动机是一种简单的状态机,其中状态的转移和具名的动作(named action)相联. 动作被分类为输入、输出或内部的. 输入和输出用于和自动机的环境交流,内部动作只为自动机自己可见. 假定输入动作不受自动机的控制 —— 它们只是从外部到来,而自动机自己来决定完成什么输出和内部动作.

I/O 自动机的一个典型例子是异步分布式系统中的一个进程. 一个典型进程自动机的接口和它的环境如图 2-1 所示.

如图 2-1(a),自动机 P_i 画成一个圆圈,入向箭头由输入动作标识,出向箭头由输出动作标识,内部动作不画出来. 所画的自动机从外界接收形如 $\mathrm{init}(v)_i$ 的输入,它表示输入值 v 的接收. 它送出形如 $\mathrm{decidev}(v)_i$ 的输出,表示基于 v 的一个判断. 为了达到一个判断,进程 P_i 可能需要通过一个消息系统和其它进程通信. 它对消息系统的接口由形如 $\mathrm{send}(m)_{i,j}$ 的输出和形如 $\mathrm{receive}(m)_{j,i}$ 的输入动作标识(注意下标次序),分别表示进程 P_i 送一个内容为 m 的消息给进程 P_j,进程 P_i 从进程 P_j 接收一个内容为 m 的消息. 当自动机做任何上述指定动作时(或任何内部动作),它也可能改变自己的状态.



(a) 进程 I/O 自动机　　(b) 通道 I/O 自动机

图 2-1

典型 I/O 自动机的另一个例子是 FIFO 消息通道.图 2-1(b) 也画出了一个典型的通道 I/O 自动机.它的输入动作形如 $send(m)_{i,j}$,输出动作形如 $receive(m)_{i,j}$.注意,通道的send意味着它接收某个进程的发送.这种标注方式适于自动机的合成操作.通常,用 I/O 自动机描述一个分布式系统时,人们将一组进程自动机和一组通道自动机合成起来,让一个自动机的输出和另一个自动机同名的输入匹配.这样,进程 P_i 所做的 $send_{i,j}$ 输出和通道 $C_{i,j}$ 的输入 $send_{i,j}$ 连接起来.在这里,异步性体现在各种动作一次只做一个,但其次序是不可预知的.这和同步系统不同,在那里所有进程依照计算过程的轮次,同时发送消息,然后同时接收消息.

形式上讲,I/O 自动机首先要说明的是其特征要素(signature),它是其输入、输出和内部动作的一种描述.假设存在那么一个动作的总体集合,特征要素 S 是一个含三个不相交动作集合的三元组:输入动作 in(S),输出动作 out(S),以及内部动作 int(S).定义外部动作 ext(S) 为 in(S) ∪ out(S);由本地控制的动作 local(S) 为 out(S) ∪ int(S);acts(S) 为所有动作.外部特征要素 extsig(S) 定义为(in(S), out(S),∅).人们经常称外部特征要素为外部接口.有了这些术语的准备,I/O 自动机可定义如下.

I/O 自动机 A,有时也简单地称为自动机,由五个部分构成:

- sig(A)　一个特征要素.
- states(A)　一个状态集,不一定有限.
- start(A)　states(A) 的一个非空子集,称为起始状态或初始状态.
- trans(A)　状态转移关系,trans(A) 包含于 states(A) × acts(sig(A)) × states(A) 之中.具有性质:对每一状态 s 和输入动作 π,都有一个转移(s,π,s')属于 trans(A).
- tasks(A)　任务划分,local(sig(A)) 上的一个等价关系,最多有可数个等价类.

用 acts(A) 来简单表示 acts(sig(A)),类似有 in(A),等等.若自动机没有输入,即 in(A) = 0,则称 A 是封闭的.

这个定义看起来和同步网络模型中的进程相似.然而,特征要素允许更一般的动作,而不只是如同同步情形的消息的发送和接收.如同同步网络模型中的进程状态集合一样,这里的状态集也不一定是有限的.这种一般性是重要的,因为它使我们能够为有无界数据结构的系统建立模型,例如计数器和无限长队列.也和同步模型一样,允许多个起始状态,这样就能将某些输入信息包含在起始状态中.

称 trans(A)的一个元素(s,π,s')为 A 的一个转移或 A 执行的一步,取决于 π 是输入动作还是输出动作,转移(s,π,s')被相应地称为输入转移或输出转移.和同步模型不同,转移不一定非得和消息的接收联系在一起,它们可以和任意动作相联.

如果对于一个特别的状态 s 和动作 π,A 有某个形如(s,π,s')的转移,则称 π 在 s 上有效(即 π 可以引起自动机从 s 状态转移到另一个状态).由于定义中特别要求每个输入动作都要在每个状态上有效,自动机也就说成是输入见效的(input enabled).输入见效假设意味着自动机不能阻塞输入动作的发生,即当消息到来时,一个进程必须要有准备对付任何消息的值.如果状态 s 只是当输入动作发生时才发生转移,则说状态 s 是休眠的.

输入见效性质有两个好处.首先,在开发系统部件中,一个严重的错误来源是不能指明在面对意外输入时部件做什么事情.用一个要求考虑任意输入的模型,对消除这种错误有帮助.第二,输入见效要求使本模型的基本理论能够很好地成为一体;有些重要的定理只是在这个性质下才能成立.

I/O 自动机定义的第五个成分,任务划分 tasts(A),可看作是自动机内"任务"或者"控制线索"的抽象描述.这个划分用来定义自动机执行的公平性条件.这种条件使得自动机在它的执行期间给与它的任务以公平的机会向前推进.对于那些从事多项工作的系统部件,这是有用的.例如,某个进程一方面参与一个正在进行的算法,同时周期性地向环境报告状态信息.另一方面,当多个自动机合成起来成为一个较大的自动机,以表示某个完整系统时,这也是有用的,任务划分此时用来说明被合成的自动机都继续在合成的系统中推进.任务划分的另一个用途是为异步共享存储算法建立模型.通常一个任务划分类被简单地称为任务.如果任务 C 的某个动作在 s 上见效,有时也称任务 C 在状态 s 上见效.

下面给出两个简单 I/O 自动机的例子.在关于 I/O 自动机的大多数描述中,转移关系的描述表现为一种"先决条件-效果"(precondition-effect)的风格.这种风格将所有涉及每个特别类型的动作的转移组织在一个代码块中,通过前置状态 s 上的一个谓词,来说明动作允许发生的条件.然后它描述由于动作发生所产生的变化.这个描述形如一个简单的程序,应用于 s 而生成 s'.作为单个转移,整块代码不可分地得到执行.因为涉及每个动作的转移通常只涉及一小部分状态,将转移按照它们的动作组织在一起有利于产生简明的代码.

写成"先决条件-效果"风格的程序一般只用到很简单的控制结构.这便于从程序到 I/O 自动机的转换,也使得关于自动机的形式化推理比较容易.

例 1　通道 I/O 自动机.

作为 I/O 自动机的一个例子,考虑一个表示通信的通道自动机 $C_{i,j}$.设 M 为一固定的消息字符集.首先给出特征要素 sig($C_{i,j}$).下面将用如下惯例:如果不提到

特征要素的某个组成部分(通常,内部动作),那么相应的动作集就是空.

特征要素:

输入:

$\text{send}(m)_{i,j}, m \in M$,

输出:

$\text{receive}(m)_{i,j}, m \in M$.

和同步系统的情形一样,描述状态 states($C_{i,j}$) 和起始状态 start($C_{i,j}$) 最方便的方式是用一个状态变量表和它们的初值.

状态:

queue,一个 M 中元素的 FIFO 队列,初始为空.

$C_{i,j}$ 的转移由下面的代码表达:

转移:

$\text{send}(m)_{i,j}$

　　效果:

　　　　将 m 加到 queue 里

$\text{receive}(m)_{i,j}$

　　先决条件:

　　　　m 是 queue 的第一个元素

　　效果:

　　　　移走 queue 中的第一个元素

这段代码应该是自我解释得很清楚的:send 动作允许在任何时候发生(即通道是无限长的),且有将消息加到队列末尾的效果,而 receive 动作只是当所论消息处于队列的头部才能发生,效果是将它移走.

任务划分 tasks($C_{i,j}$),将所有 receive 动作收集到一个任务中,即接收消息(对通道来说是向外投递)的工作被看作是一项任务.

任务:

$\{\text{receive}(m)_{i,j} \mid m \in M\}$

例 2　进程 I/O 自动机.

作为 I/O 自动机的第二个例子,考虑进程自动机 P_i.这个自动机有如下所述的外部接口.这里 V 是一个固定的值的集合,null 是一个特殊的不在 V 中的值,f 是一个函数:$f: V^n \to V$.

特征要素:

输入:

$\text{init}(v)_i, v \in V$

$\text{receive}(v)_{j,i}, v \in V, 1 \leq j \leq n, j \neq i$

输出:

$\text{decide}(v)_i, v \in V$

$\text{send}(v)_{i,j}, v \in V, 1 \leq j \leq n, j \neq i$

状态:

val,一个由 $\{1,2,\cdots,n\}$ 索引的向量,其中的元素属于 $V \cup \{\text{null}\}$,初值均为 null.

转移:

$\text{init}(v)_i, v \in V$

　　效果:

　　　　$\text{val}(i) := v$

$\text{receive}(v)_{j,i}, v \in V$

　　效果:

　　　　$\text{val}(j) := v$

$\text{send}(v)_{i,j}, v \in V$

$\text{decide}(v)_i, v \in V$

先决条件:
　　$\text{val}(i) = v$
效果:
　　none

先决条件:
　　for all $j, 1 \leqslant j \leqslant n$
　　$\text{val}(j) \neq \text{null}$
　　$v' = f(\text{val}(1), \cdots, \text{val}(n))$
效果:
　　none

这样,初始动作引起 P_i 将给定的值填充在 val 向量的相应位置,而 receive 动作引起它填充另外的位置.多次 init 和 receive 动作使这些值能被更新任意多次.P_i 被允许在任何通道上发送它自己的值任意多次.基于 f 在它的向量上的施行结果,P_i 还被允许做任意次判断.

任务划分 tasks(P_i),包含 n 个任务:所有 $\text{send}_{i,j}, j \neq i$,对应一任务,所有 decide 动作也对应一个任务.这样,在一个通道上的发送被看作是一个任务,报告判断也是如此.

任务:
for every $j \neq i$
　　$\{\text{send}(v)_{i,j} \mid v \in V\}$
$\{\text{decide}(v)_i \mid v \in V\}$

现在描述 I/O 自动机 A 的执行机制.A 的一个执行片段(execution fragment)或者是一个有穷序列 $s_0, \pi_1, s_1, \pi_2, \cdots, \pi_r, s_r$,或者是一个无限序列 $s_0, \pi_1, s_1, \pi_2, \cdots, \pi_r, s_r, \cdots$,其中 A 的状态和动作交替出现,且对所有 $k \geqslant 0$,序列中的$(s_k, \pi_{k+1}, s_{k+1})$ 是 A 的一个转移.注意,如果序列是有穷的,则它必须以状态结尾.用一个初始状态开头的执行片段称为一次执行.用 exec(A) 来代表 A 的执行的集合.A 的一个状态称为是可达的(reachable),如果它是 A 的某个有限执行的终止状态.

如果 α 是 A 的一个有限执行片段,α' 是从 α 的最后一个状态开始的另一个执行片段,那么记 $\alpha \cdot \alpha'$ 为它们的拼接(除去交界处因重复多出的那个状态).显然,$\alpha \cdot \alpha'$ 也是 A 的一个执行片段.

有时只观察一个 I/O 自动机的外部行为.这样,A 的一次执行 α 的轨迹,记做 trace(α),是 α 的包含外部动作的子序列.如果 β 是 A 的某个执行的轨迹,则称 β 是 A 的一个轨迹.A 的所有轨迹记做 traces(A).

例 3 执行.

下面是例子 1 所描述的自动机 $C_{i,j}$ 的 3 个执行(假定消息字符集 M 等于集合 $\{1,2\}$),其中状态由将队列中的字符序列放在方括弧中来表示(λ 表示空序列).

　　$[\lambda], \text{send}(1)_{i,j}, [1], \text{receive}(1)_{i,j}, [\lambda], \text{send}(2)_{i,j}, [2], \text{receive}(2)_{i,j}, [\lambda]$
　　$[\lambda], \text{send}(1)_{i,j}, [1], \text{receive}(1)_{i,j}, [\lambda], \text{send}(2)_{i,j}, [2]$
　　$[\lambda], \text{send}(1)_{i,j}, [1], \text{send}(1)_{i,j}, [11], \text{send}(1)_{i,j}, [111], \cdots$

注意,后面两个执行尽管含有发送了但不会被收到的消息,这种情况是允许的.这是因为到目前为止对执行没有任何限制,没有规定见效的动作必须出现.

如上所定义的 I/O 自动机可以用来对应分布式系统的最基本组成部分,为了建立一个复杂分布式系统的模型,有可能需要将多个这样的自动机连接起来.这通

过所谓I/O自动机的合成操作(composition)来实现.合成操作的基本思路是将代表不同部件的自动机中相同名字的动作等同起来.当任何部件自动机完成涉及动作π的一步,所有在其特征要素中有π的部件自动机也都完成一步.

我们给参与合成的自动机强加某些限制.首先,由于自动机A的内部动作意在不被任何其它自动机B观察到,如果A的内部动作和B的动作有相交,则不允许A和B合成(否则,A进行一个内部动作要引起B推进).再者,为了使合成操作可能满足一些好的性质,建立一个惯例:最多一个部件自动机"控制"任何给定动作的进行.这样,如果A和B的输出动作交集不为空,则不允许A和B的合成.第三,不排除合成无穷但可数个自动机的可能性,但要求在这种情况下每个动作必须只是有限个自动机的动作.

为什么我们不简单地排除合成无限多个自动机的可能性?说到底,实际的计算机系统只是由有限多个部件(计算机,消息通道等)构成的,这里的理由是:I/O自动机可以用来既为物理系统建立模型,也为逻辑系统建立模型.一个逻辑的系统可能由大量的逻辑部件构成,但却在一个只有较少部件的物理系统上实现.事实上,某些逻辑系统允许部件在执行期间动态创建(例如,数据库系统能允许系统在执行时创建新的交易实例).用I/O自动机为部件的动态创建建立模型的方式是想像所有可能被创建的部件实际上从一开始就存在,但需特别的唤醒动作,在它们假定被创建时唤醒.用这种建模的技巧,普通合成操作就足以来描述动态创建的部件和系统中其它部分的交互方式.但这就有必要允许无穷多部件的组合.

形式化地讲,一个特征要素的可数集合$\{S_i\}_{i\in I}$称为相容的,如果对所有i、$j\in I$,$i\neq j$,下面都成立:

1° $\mathrm{int}(S_i)\cap \mathrm{acts}(S_j)=\varnothing$;

2° $\mathrm{out}(S_i)\cap \mathrm{out}(S_j)=\varnothing$;

3° 没有动作包含在无限多个集合$\mathrm{acts}(S_i)$中.

称一个自动机集合是相容的,如果其中自动机的特征要素是相容的.

当将若干自动机合成起来时,那些自动机称为合成的部件,部件的输出动作成为合成的输出动作,部件的内部动作成为合成的内部动作,那些是某些部件的输入动作但不是任何部件的输出动作成为合成的输入动作.上述这些描述的形式化表示为:可数个相容特征要素$\{S_i\}_{i\in I}$的合成$S=\prod_{i\in I}S_i$具有特征要素

- $\mathrm{out}(S)=\bigcup_{i\in I}\mathrm{out}(S_i)$
- $\mathrm{int}(S)=\bigcup_{i\in I}\mathrm{int}(S_i)$
- $\mathrm{in}(S)=\bigcup_{i\in I}\mathrm{in}(S_i)-\bigcup_{i\in I}\mathrm{out}(S_i)$

现在可以定义可数个相容的I/O自动机$\{A_i\}(i\in I)$的合成$A=\prod_{i\in I}A_i$了.它是如下定义的自动机.

- $\mathrm{sig}(A)=\prod_{i\in I}\mathrm{sig}(A_i)$,

- $\text{states}(A) = \prod_{i \in I} \text{states}(A_i)$,
- $\text{start}(A) = \prod_{i \in I} \text{start}(A_i)$,
- trans(A)是元素形如三元组(s,π,s')的集合.其中对所有$i \in I$,如果$\pi \in \text{acts}(A_i)$,则$(s_i,\pi,s'_i) \in \text{trans}(A_i)$;否则$s_i = s'_i$.
- $\text{tasks}(A) = \bigcup_{i \in I} \text{tasks}(A_i)$.

这样,合成自动机的状态和起始状态分别是部件自动机状态和起始状态的向量.上述合成的转移的定义讲的是,所有含有动作π的部件自动机同时参与合成自动机中涉及到π的执行步,而其它部件自动机什么也不做.合成的局部控制动作的任务划分由部件任务划分的并形成,即每个部件自动机的每个等价类成为合成的一个等价类.这意味着,当部件被合成时,单个部件的任务结构被保持.注意,由于自动机A_i是输入见效的,它们的合成也是如此.综合起来,这意味着合成也是一个I/O自动机.

当I是有限集时,有时用中位操作符"×"来表示合成.例如,如果$I = \{1,2,\cdots,n\}$,有时记$A = \prod_{i \in I} A_i$为$A_1 \times A_2 \times \cdots \times A_n$.

注意,如果一个部件的输出动作π还是另一个部件的输入,那么它被归类为合成的输出动作,而不是内部动作.这是由于要允许用π作进一步通信的可能性.例如,假设自动机A有输出动作π,而自动机B和C都有输入动作π,这样,π在合成$A \times B \times C$中就是一个从A到B和C的广播动作.希望能用一种模块化的方式来考虑这种合成,首先构造$A \times B$,然后将结果与C合成.按照定义合成的方式,$A \times B \times C$和$(A \times B) \times C$实际上是同构的.但如果π在$A \times B$中被分类为内部的,那么就不能有这种模块性:合成$A \times B$甚至不能和C再合成,这是因为第一相容条件不再满足.

例4 自动机的合成

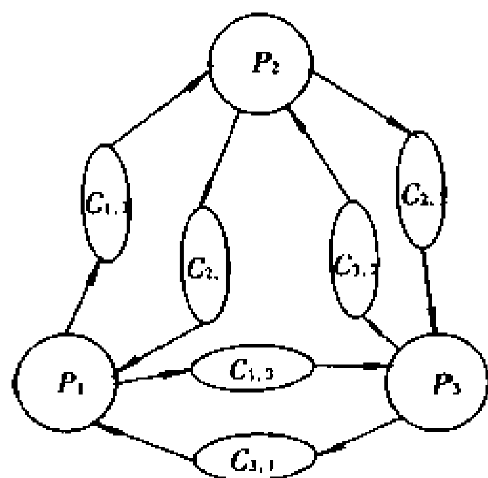


图 2-2

考虑索引集合$I = \{1,2,\cdots,n\}$,让A是例2中所有进程自动机$P_i(i \in I)$和例1中所有通道自动机$C_{i,j}(i、j \in I)$的合成.为了合成它们,必须假定通道自动机的消息字符集M包含进程自动机的值集V.图2-2示意了当$n = 3$时进程自动机和通道自动机的合成.合成结果是代表一个分布式系统的单个自动机.系统的状态:每个进程一个状态(值向量),加上每个通道一个状态(消息队列).系统的每个转移涉及如下情况之一:

(1) $\text{init}(v)_i$输入动作,给进程P_i的$\text{val}(i)$赋一个值,$\text{val}(i)_i$.

(2) $\text{send}(v)_{i,j}$输出动作,进程P_i的值$\text{val}(i)_i$

被送给通道 $C_{i,j}$.

(3) $\text{receive}(v)_{i,j}$ 输出动作，通道 $C_{i,j}$ 的第一个消息被移去，同时放到进程 P_j 的变量 $\text{val}(i)_j$ 中.

(4) $\text{decide}(v)_i$ 输出动作，进程 P_i 通报它当前算得的值.

当 $n = 2$，设 V 是整数 N、f 是加操作，可得这个合成的一个轨迹如下：

$\text{init}(2)_1$, $\text{init}(1)_2$, $\text{send}(2)_{1,2}$, $\text{receive}(2)_{1,2}$, $\text{send}(1)_{2,1}$,

$\text{receive}(1)_{2,1}$, $\text{init}(4)_1$, $\text{init}(0)_2$, $\text{decide}(5)_1$, $\text{decide}(2)_2$.

在从这个轨迹能到达的唯一的系统状态里，P_1 有 val 向量(4,1)，P_2 有 val 向量(2,0)，两个通道都为空. 当然，这个合成系统的执行还可能有许多其它轨迹.

可以看到，如上所定义的 I/O 自动机模型是一个相当严密的形式化系统，因此有可能导出一些有用的性质. 例如可以将一个合成的执行和轨迹与其部件自动机的执行和轨迹联系起来，下面就是这样的 3 个结果. 第一个结果讲合成的一个执行或轨迹能"投影"产生部件自动机的执行和轨迹. 给定 A 的一个执行，$\alpha = s_0, \pi_1, s_1, \cdots$ 让 $\alpha \mid A_i$ 表示从 α 中删除和 A_i 的动作无关的 π_r、s_r，并用 $(s_r)_i$ 替代剩下的 s_r 所得到的序列（$(s_r)_i$ 即为状态 s_r 的 A_i 部分）. 另外，给定 A 的一个轨迹 β（更一般地，是任何一个动作序列），让 $\beta \mid A_i$ 表示 β 中只含 A_i 的动作的子序列.

定理 1 设 $\{A_i\}_{i\in I}$ 为相容自动机的一个集合，且 $A = \prod_{i\in I} A_i$.

1° 如果 $\alpha \in \text{execs}(A)$，则对每一个 $i \in I$，$\alpha \mid A_i \in \text{execs}(A_i)$.

2° 如果 $\beta \in \text{traces}(A)$，则对每一个 $i \in I$，$\beta \mid A_i \in \text{traces}(A_i)$.

另外两个是这个定理的逆定理. 下一个定理讲在一定的条件下，部件自动机的执行能够"粘贴起来"形成合成的一个执行.

定理 2 设 $\{A_i\}$，$i \in I$ 为相容自动机的一个集合，且 $A = \prod_{i\in I} A_i$. 对每一个 $i \in I$，如果 α_i 是 A_i 的一个执行，且对于 $\text{ext}(A)$ 中的动作序列 β，有 $\beta \mid A_i = \text{trace}(\alpha_i)$，则存在 A 的一个执行 α，满足 $\beta = \text{trace}(\alpha)$ 且对于每一个 $i = I$，$\alpha_i = \alpha \mid A_i$.

最后一个定理讲的是部件自动机的轨迹也能粘贴起来形成合成的轨迹.

定理 3 设 $\{A_i\}$，$i \in I$ 为相容自动机的一个集合，且 $A = \prod_{i\in I} A_i$. 假定 β 是 $\text{ext}(A)$ 中的一个动作序列. 如果对每一个 $i \in I$，$\beta \mid A_i \in \text{traces}(A_i)$，则 $\beta \in \text{traces}(A)$.

这个定理告诉我们，为了说明一个序列是某个系统的轨迹，只要说明它在每单个系统部件上的投影是部件的轨迹就够了.

在分布式系统中，通常只对那些所有部件都能得到公平的机会的执行感兴趣. 这里，定义适用于 I/O 自动机的一种公平性概念.

如前所述，每个 I/O 自动机有一个它局部控制动作的划分；其中的每个等价类代表自动机要完成的某个任务. 这里的公平性概念旨在要求每个任务都能得到无穷多的执行机会.

形式化地讲，若下列条件对 tasks(A) 中的每个类 C 成立，则称 I/O 自动机 A 的执行片段 α 是公平的.

1° 若 α 是有限的，则 C 不在 α 的终结状态上有效.

2° 若 α 是无穷的，则 α 要么包含来自于 C 的无穷多个事件，或无穷多个 C 在其上无效的状态.

这里，用事件这个词来代表某个动作在一个序列中的出现.

可以理解这个定义的含义即是要每个任务(等价类) C 都得到无穷多次执行的机会.只要满足这个条件，要么 C 的某个动作得到执行，或者由于没有动作是有效的，因此 C 中不可能有动作会得到执行.对于有限执行片段的情形，可以认为在一个公平执行结束时，自动机以一种轮流的方式不断给与每个任务以执行机会，但由于没有动作在终结状态上是有效的，于是没导致任何动作的完成.

用 fairexecs(A) 表示 A 的公平执行的集合.如果 β 是 A 的某个公平执行的轨迹，则称 β 为 A 的一个公平轨迹.

以例 3 中的三个执行为例，第一个执行是公平的，第二个是不公平的，第三个也是不公平的.这是因为在第一个执行中没有 receive 动作在它的终结状态上有效；第二个为一有限执行片段，receive 动作在其终结状态上是有效的；第三个为无穷执行片段，不包含 receive 事件，但在第一步后的每一点 receive 都是有效的.

2.1.3 异步共享存储模型

如前面所提到过的，"同步／异步"和"共享存储／消息传递"是区别分布式系统模型的两大正交的特性.同样，在开始介绍 I/O 自动机模型时提到过，I/O 自动机是描述各种异步系统的一个基础性模型.因此，本节的目的有两个，一是介绍一种异步共享存储模型，二是介绍 I/O 自动机的一个应用.换句话说，我们要讨论的是用 I/O 自动机来建立一个异步共享存储系统模型.特别要指出的是，建立异步共享存储系统的模型，不一定非得用 I/O 自动机；同样是用 I/O 自动机，模型也可能建立得不一样.这里提供的只是一个实例.

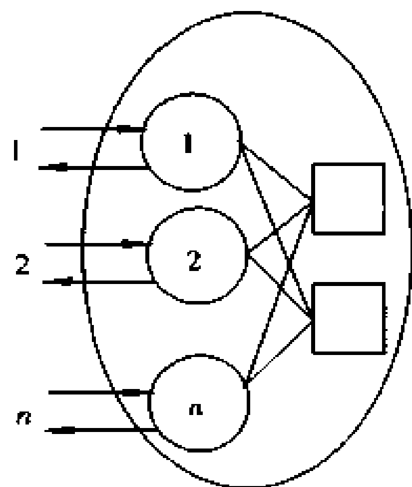


图 2-3

如同网络系统一样，共享存储系统也是由一组相互通信的进程构成.但此时进程的通信通过对共享变量操作来实现，而不是通过通信通道发送和接收消息.一般来讲，异步共享存储系统由一个进程的有穷集合构成，它们通过一个有穷共享变量集合来相互作用.这些变量进程内部可能有其它状态变量，只是用来支持系统中进程的通信.另外，由于外部世界应该能和共享存储系统相互作用，故还假定每个进程有一个端口(port)，通过它用输入输出动作和外部世界交流.典型的情形如图 2-3 所示(异步共享存储系统示意图).

用 I/O 自动机建立共享存储系统的模型，一种看起来很自然的方法是让每个进程对应一个自动机，让每个共享变量也对应一个自动机，然后再考虑某种形式的合成.但那将会导致某些复杂的情形.这里的做法是直接用一个"大"I/O

自动机 A 来作为整个共享存储系统的模型．这里的任务有两个方面，一是针对这个系统给出 I/O 自动机定义的五个要素，二是要表达出系统的结构信息．对于后者，这里的做法是通过给系统的事件加上某些局部性限制来实现．

如同同步网络系统，用 $1,2,\cdots,n$ 来标记系统中的进程．设每个进程 i 有一个相关的状态集 $states_i$，其中某些被认定为起始状态 $start_i$．还设每个共享变量 x 有一个取值集合 $values_x$，其中某些被认定为初值 $initial_x$．于是，states(A) 如此定义：它的每个状态包含 $states_i$ 的一个状态（对所有 i），加上 $values_x$ 的一个值（对所有 x）．相应地，start(A) 的每个状态包含 $start_i$ 的一个状态（对所有 i），加上 $initial_x$ 的一个值（对所有 x）．

让 acts(A) 的每个动作都和某个进程联系在一起．除此以外，int(A) 中的某些内部动作可能和某个共享变量联系起来．和进程 i 相联的输入和输出动作被用来对应进程 i 和外部世界的相互作用，称它们发生在端口 i．进程 i 的内部动作如果和共享变量没什么关系，则用来参与局部计算；若和共享变量 x 有关系，则用来完成在 x 上的操作．

为了刻画系统的进程和共享变量形成的结构，给转移的集合 trans(A) 加上一些局部性限制．首先考虑一个和进程 i 相关，但不和任何变量相关的动作 π．那么如前所述，π 用来做局部计算，它只影响进程 i 的状态．于是，π 的转移集合就能够从元素形如 (s,π,s') 的某个集合产生，其中 s、$s' \in states_i$．产生的方式可以是给 s 和 s' 附着其它进程状态和共享变量的任意组合．

另一方面，考虑一个既和进程 i 相关，也和某个变量 x 相关的动作 π；如前所述，进程 i 用 π 来在 x 上施行某种操作．那么只是 i 的状态和 x 的值和 π 有关．于是，π 的转移集合就能够从元素形如 $((s,v),\pi,(s',v'))$ 的某个集合产生，其中 s、$s' \in states_i$，$v,v' \in values_x$．产生的方式可以是给 s 和 s' 附着其它进程状态和其它共享变量的任意组合．这里有一个技术细节：如果 π 和进程 i 与变量 x 相关，π 是否产生效果只取决于进程 i 的状态．x 的值只影响所产生的变化．也就是说，π 在进程 i 处于状态 s 和变量 x 取值 v 时有效，它对于 s 和 x 的另一取值 v' 也是有效的．

任务的划分 tasks(A)，必须和进程结构一致：即每个等价类（任务）应该只包含一个进程的局部控制动作．下面看一个例子．

例 5 共享存储系统．

设 V 为一个值的集合．考虑含有 n 个进程和一个共享变量 x 的共享存储系统 A．进程用 $1,2,\cdots,n$ 编号，x 取值 $V \cup \{unknown\}$，初始为 unknown，．输入形如 $init(v)_i$，其中 $v \in V$，i 为进程标识．输出形如 $decide(v)_i$．内部动作形如 $access_i$．所有带有下标 i 的动作和进程 i 相关．除此以外，access 动作和变量 x 相关．

当进程 i 接收到一个 $init(v)_i$ 输入，它访问 x．如果它发现 x = unknown，它就将其 v 值写进 x 并形成 v 上的一个判断．如果 $x = w, w \in V$，那么它就不向 x 写任何东西，而形成在 w 上的一个判断．

这里，每个集合 $states_i$ 由局部变量组成．

states of i：

status $\in$ {idle, access, decide, done}，初始为 idle

input ∈ V ∪ {unknown},初始为 unknown
output ∈ V ∪ {unknown},初始为 unknown
Transitions of i:

init(v)$_i$
  效果:
    input: = v
    if status = idle then
      status: = access

decide(v)$_i$
  先决条件:
    status = decide
    output = v
  效果:
    status: = done

access:
  先决条件:
    status = access
  效果:
    if x = unknown then x: = input
    output: = x
    status: = decide

每个进程有一个任务,它包含其所有 access 和 decide 动作.

不难看到,在 A 的每一个公平执行中,任何接收了一个 init 输入的进程最终总会做 decide 输出.进而,每个执行都满足所谓"同一性质",即没有两个进程在不同的值上作判断,还有"有效性性质",即每个判断的值都是某个进程的初值.

2.1.4 部分同步模型

在同步模型和异步模型之间,人们还研究了所谓**部分同步模型**(partial synchronous system).在部分同步系统中,每个元素有一部分关于时间的信息,但这些信息可能并不精确.例如,一个部分同步系统中的进程可以访问准同步的时钟,可以知道所有进程执行步长时间或消息递交时间的上界等等.部分同步系统很可能比完全的同步系统和完全的异步系统更接近于实际系统,因为实际系统往往拥有部分时间信息.但是部分同步系统理论不像同步系统和异步系统那样成熟和完善,算法也不多,所以只在此简单介绍两个比较公认的部分同步系统的模型:MMT 时限自动机(MMT Timed Automaton)模型和通用时限自动机(General Timed Automation,GTA)模型.其中,MMT 是 GTA 的一种特例,可以映射到 GTA.MMT 是该模型三个创立者名字的头一个字母.

1. MMT 时限自动机模型

MMT 时限自动机模型是将 I/O 自动机中的公平性,用时间的上界和下界代替而得到的.MMT 时限自动机 $B = (A, b)$,A 是一个只有有限多个任务的 I/O 自动机,b 是 A 的界限映像(boundmap).界限映像规定所有任务的下界和上界由两个映像(lower 和 upper)组成.对每个任务 C 而言,界限映像必须满足下列条件:

$$0 \leqslant \text{lower}(c) < \infty, \quad 0 < \text{upper}(c) \leqslant \infty, \quad \text{lower}(c) \leqslant \text{upper}(c).$$

MMT 自动机 $B = (A, b)$ 的时限执行(timed execution)定义为一个有穷序列:

$$\alpha = s_0,(\pi_1,t_1),s_1,(\pi_2,t_2),\cdots,(\pi_r,t_r),s_r$$

或者一个无限序列：

$$\alpha = s_0,(\pi_1,t_1),s_1,(\pi_2,t_2),\cdots,(\pi_r,t_r),s_r,\cdots,$$

其中：s 是 I/O 自动机 A 的状态，π 是 A 的动作，t 是非负实数的时间($t \in \mathbf{R} \geq 0$)。规定：$s_0,\pi_1,s_1\cdots$ 是 A 的一个执行，α 中的连续时间 t_r 不减小并且满足界限映像 b 规定的上界和下界。B 的时限执行集合表示为 texecs(B)。如果一个状态是 B 的某些时限执行的最终状态，则称该状态可达(reachable)。

一个任务 C 的初始索引(initial index) r 定义为：

C 在 S_r 状态时是激活的，并且下述条件之一成立：

1° $r = 0$;

2° C 在 s_{r-1} 时未被激活；

3° $\pi_r \in C$。

初始索引表示开始度量时限的时刻。

一个任务 C 被称为满足界限映像，如果 C 的所有初始索引满足下述两个条件：

- 上界：如果存在 $k > r, t_k > t_r + \text{upper}(C)$，则必存在 $k' > r, t_{k'} < t_r + \text{upper}(C), \pi_{k'} \in C$ 或者 C 在 $S_{k'}$ 未被激活。
- 下界：不存在 $k > r, t_k < t_r + \text{lower}(C)$ 并且 $\pi_k \in C$。

如果一个时限执行满足下述条件，则称其为可接受的：

- 可接受性：如果 α 是无限序列，则动作的时间趋于 ∞；如果 α 是有限序列，那么在 α 的最终状态激活的任务 C，有 $\text{upper}(C) = \infty$（即如果自动机有更多的工作要做时，进程不会停止）。

B 的可接受时限执行集合表示为 atexecs(B)。

上界和下界是 B 的安全性质(safety properties)，可接受性是 B 的基本活跃特性(liveness property)。

为了描述 MMT 自动机的外部行为，引入时限轨迹(Timed Traces)。B 的一个时限执行 α 的时限轨迹是包含所有 B 的外部动作(每个动作有一个关联时间)的 α 的子序列，表示为 ttrace(α)。B 的可接受时限轨迹是 B 的可接受时限执行的时限轨迹，表示为 attrace(B)。

MMT 自动机模型可用来描述部分同步系统中的很多元素，尤其适用于为低层的计算机系统建立模型，但是不太适用于描述高层的计算机系统。

2. 通用时限自动机模型

MMT 自动机模型用上界和下界对执行的时间进行限制，通用时限自动机模型则把时间限制与自动机的状态及状态转移直接联系起来，这种方法的优点是便于使用基于状态证明方法（如不变断言法和模拟关系法）证明时限系统的正确性和时间特性。

为了定义 GTA 模型，首先区别几类动作：

时间段动作(time-passage actions) $v(t)$, $t \in \mathbf{R}^+$：表示长度为 t 的时间段内的动作；

时限特征要素(timed signature) S：是一个由输入动作 in(s)、输出动作 out(s)、内部动作 int(s) 和时间段动作等四组不同的动作集合组成的四元组. 定义：

- 可见动作(visible actions) vis(s)：vis(s) = in(s) $\cup$ out(s).
- 外部动作(external actions) ext(s) = vis(s) $\cup$ $\{v(t) \mid t \in \mathbf{R}^+\}$.
- 离散动作(discrete actions) disc(s)：= vis(s) $\cup$ int(s).
- 本地控制动作(locally controlled actions) local(s)：= out(s) $\cup$ int(s).
- acts(S)：S 的所有动作.

一个 GTA A 由四部分组成：

- sig(A)：一个时限特征要素.
- states(A)：一组状态.
- start(A)：一个起始状态或初始状态集，是 states(A) 的非空子集.
- trans(A)：一个状态转移关系，是 states(A) × acts(sig(A)) × states(A) 的子集.

注意：与 I/O 自动机和 MMT 自动机不同，GTA 没有 tasks(A) 元素. 另外和以前一样，act(sig(A)) 简写为 acts(A)，in(sig(A)) 简写为 in(A).

A 必须满足下述两个简单的公理：

A_1：如果(S, $V(t)$, S') 和(S', $V(t')$, S'') 在 trans(A) 中，则(S, $V(t+t')$, S'') 亦在 trans(A) 中

A_2：如果(S, $V(t)$, S') $\in$ trans(A)，并且 $0 < t' < t$，则存在一个状态 S''，(S, $V(t')$, S'') 和(S'', $V(t-t')$, S') 都在 trans(A) 中.

GTA A 的一个时限执行片段(timed execution fragment) 定义为一个有限序列：

$$\alpha = S_0, \pi_1, S_1, \pi_2, \cdots, \pi_r, S_r,$$

或者一个无限序列：

$$\alpha = S_0, \pi_1, S_1, \pi_2, \cdots, \pi_r, S_r, \cdots,$$

其中 S 是 A 的状态，π 是 A 的动作(可为输入动作，输出动作，内部动作或时间段动作)，(S_k, π_{k+1}, S_{k+1}) 是 A 的转移，以一个起始状态开始的时限执行片段称为一个时限执行.

如果 α 是任何时限执行片段，π_r 是 α 中的任何离散动作，那么称 π_r 的出现时间是 α 中 π_r 前所有时间段动作的时间值之和. 如果 α 中所有时间段动作的时间之和为 ∞，则称 α 为可接受的时限执行片段. A 的所有可接受时限执行的集合表示为 atexecs(A). 我们主要考虑可接受时限执行，偶尔考虑有限序列的时限执行. 如果 A 中的一个状态是某个有限时限执行的最终状态，则称该状态是可达的.

一个时限执行片段 α 的时限轨迹是 α 中可见事件的序列(每个事件带有它的

出现时间). A 的可接受时限轨迹是 A 的可接受时限执行的时限轨迹,表示为 attrace(A).无限的可接受时限执行的 attrace(A) 可能是有限的.

如果两个时限执行片段 α、α',除了 α 中的某些时间段步骤可以用α' 中有限的时间段步骤序列替换而保持相同的起始状态、最终状态及时间段总长度不变之外完全一样,则称 α 是α' 的时间段求精(time-passage refinement).如果两个时限执行片段有相同的时间段求精,则称 α 和α' 时间段相等(time-passage equivalent).

部分同步系统和算法的正确性和性能在很大程度上依赖于时间假设,任何时间上小的变化都可能极大地改变系统的行为.为了分析部分同步系统对时间的依赖关系,必须使用系统的证明方法.和同步系统、异步系统一样,证明的基本方法仍然是不变断言法和模拟关系法,但需要做一些适应性处理,并要引入一些新特性,如时限轨迹特性(timed trace property) 等等.

2.2 典型算法

在介绍了与模型和方法有关的概念后,本节给出一些典型的算法,旨在使读者能对分布式计算领域的问题描述和算法设计和分析风格有些感性认识.选择同步网络作为算法的模型.所要讨论的算法分两类,针对两个不同的典型问题:领袖推选(leader election) 问题和一致性问题.

2.2.1 同步网络中的领袖推选算法

算法一 LCR 算法

问题描述:在一个 n 个节点(顺时针方向标识为$0,1,\cdots,n-1$)的单向、环形网络中,每个节点沿顺时针方向传递消息.每个节点都有一个与其它节点不同的标识符 UID(UID 属于一个有序集),没有节点事先知道其它节点的 UID,也不知道环上有多少个节点,但每个节点可以根据相对位置识别顺时针方向的邻居和逆时针方向的邻居.节点可以对 UID 进行比较操作.要求解的问题是:n 个节点最终推选出一个唯一的领袖,该节点输出自己是领袖的判断.

算法的形式化描述:

有向图 G 由 n 个节点及其相邻节点之间的有向边(顺时针方向)组成.消息集 M 是由 UID 组成的集合.

对于每一个 i,$state_i$ 由下述变量组成:

u,一个 UID,初始值为 i 的 UID.

send,一个 UID 或者 null,初始值为 i 的 UID.

status,在集合{unknown,leader} 中取值,初始值为 unknown.

起始状态集 $start_i$ 由唯一的一个状态组成,该状态的元素值为如上定义的初始值.

对于每一个 i,消息生成函数 msg_i 定义为:

将 send 变量的值发往进程 $i+1$

/* $i+1$ 为 i 的顺时针邻居 */

```
        /* 如果 send = null,则 msg_i 实际上并不发送任何消息 */
对于每一个 i,状态转换函数 trans_i 定义为:
send: = null  /* 清除上次消息传送的结果 */
if(收到的消息是 v) then
    case
      v > u:send: = v
      v < u:什么都不做
      v = u:status: = leader
endcase
```

该算法的主要思想是:每个进程在环上发送自己的 UID.当一个进程接收到一个 UID 时,用自己的 UID 与其比较.如果该 UID 比自己的大,则继续传送该 UID;如果比自己的小,则丢弃该 UID;如果与自己的相等,该进程便宣称自己是领袖.结果是,UID 最大的进程成为唯一的领袖.

复杂度:LCR 算法的时间复杂度为 $O(n)$,通信复杂度为 $O(n^2)$.

算法二　HS 算法

问题描述:该算法的问题和 LCR 算法的问题类似,唯一的不同是 HS 算法假设环是双向的,即每个节点可以向顺时针和逆时针邻居发送消息.

算法的形式化描述:

消息集 M 是三元组(UID,flag,hop-count)的集合,其中 flag 的值取自集合{out,in},hop-count 是一个正整数.

对于每一个 i,$state_i$ 由下述变量组成:

u,一个 UID,初始值为 i 的 UID

send +:为 M 中的一个元素或者 null,初始值为(i 的 UID,out,1)

send -:为 M 中的一个元素或者 null,初始值为(i 的 UID,out,1)

status,在集合{unknown,leader}中取值,初始值为 unknown

phase:一个非负整数,初始值为 0

起始状态集 $start_i$ 由唯一的一个状态组成,该状态的元素值为如上定义的初始值.

对于每一个 i,消息生成函数 msg_i 定义为:

```
将 send + 变量的值发往进程 i + 1   /* i + 1 为 i 的顺时针邻居 */
将 send - 变量的值发往进程 i - 1   /* i - 1 为 i 的逆时针邻居 */
对于每一个 i,状态转换函数 trans_i 定义为:
  send +: = null
  send -: = null
if(从 i - 1 收到的消息是(v,out,h)) then
case
  v > u and h > 1:send +: = (v,out,h - 1)
  v > u and h = 1:send -: = (v,in,1)
  v = u:status: = leader
```

```
endcase
if(从 i + 1 收到的消息是(v,out,h)) then
case
   v > u and h > 1:send +: = (v,out,h - 1)
   v > u and h = 1:send -: = (v,in,1)
   v = u:status: = leader
endcase
if(从 i - 1 收到的消息是(v,in,1)and v ≠ u) then
     send +: = (v,in,1)
if(从 i + 1 收到的消息是(v,in,1)and v ≠ u) then
     send -: = (v,in,1)
if(从 i + 1 和 i - 1 收到的消息都是(v,in,1)) then
     phase: = phase + 1;
     send +: = (< u,out,2^phase)
     send -: = (u,out,2^phase)
```

该算法的思想是:每个进程 i 在阶段 0,1,2,… 进行操作.在每个阶段 l,进程 i 向两个方向发出包含自己 UID 的令牌.这些令牌在环上传输 2_l 步(出向),然后折向传输回 i(入向).如果两个令牌都能安全地回到原点,则进程 i 继续下一阶段.令牌 u_i 出向上的每个进程 j 比较自己的 UIDu_j,如果 $u_i < u_j$,则丢弃令牌 u_i;如果 $u_i > u_j$,则 j 传递 u_i;如果 $u_i = u_j$,则意味着进程 j 在令牌转向之前就收到了自己的 UID,所以 j 推选自己为领袖.所有进程总是传递入向的令牌.算法的结果是,UID 最大的进程成为唯一的领袖.

复杂度:HS 算法的时间复杂度为 $O(n)$,通信复杂度为 $O(n\ \mathrm{lb}n)$

算法三 FloodMax 算法

问题描述:该算法的问题和 LCR 算法的问题类似,唯一的不同是有向图 $G = (V,E)$ 不再是环形拓扑,而是任意的、强连接的网络有向图,并且每个进程都事先知道网络的直径或直径的上界.问题的要求变为:每个进程最终都要决定自己是否领袖.

算法的形式化描述:

消息集合由进程标识符组成.

```
states_i 由下述元素组成:
   u,a UID,initially i's UID
   max-uid,a UID,initially i's UID
   status ∈ {unknown,leader,non-leader},initially unknown
   rounds,an integer,initially 0

msg_i:
if rounds < diam then
     send max-uid to all j ∈ out-nbrs
```

```
trans_i:
rounds: = rounds + 1
let U be the set of UIDs that arrive from processes in in-nbrs
max-uid: = max{{max-uid} ∪ U}
if rounds = diam then
  if max-uid = u then status: = leader
  else status: = non-leader
```

该算法的主要思想是:每个进程记录到目前为止自己所看到的最大 UID(初值为自己的 UID).在每一轮,每个进程把这个最大的 UID 在自己所有的出向边上传播.diam 轮后,如果某个进程看到的最大 UID 值和自己的 UID 相等,则选举自己为领袖,否则自己为非领袖.算法的结果是 UID 值最大的进程当选为领袖.

复杂度:FloodMax 算法的时间复杂度为 diam,通信复杂度 = diam · | E |.

为了减小通信复杂性,可以改进 FloodMax 算法:发送 max-uid 值时,没有必要每轮都发送,只需要在该值第一次改变时发送即可,这种算法叫做 OptFloodMax 算法.其形式化描述可以改写为:

```
states_i 另外增加一个元素:
  new-info, a Boolean, initially true
msg_i:
if rounds < diam and new-info = true then
  send max-uid to all j ∈ out-nbrs
trans_i:
rounds: = rounds + 1
let U be the set of UIDs that arrive from processes in in-nbrs
if max(U) > max-uis then new-info: = true else new-info: = false
max-uid: = max{{max-uid} ∪ U}
if rounds = diam then
  if max-uid = u then status: = leader
  else status: = non-leader
```

OptFloodMax 的正确性可以通过证明其与 FloodMax 的模拟关系而得到证明.

2.2.2　分布式的一致性问题算法

1. 随机性的联合攻击问题(coordinated attack problem)

问题描述:在一个任意拓扑的无向图中,有 n 个编号(索引)为 $1,2,\cdots,n$ 的进程,每个进程都知道整个图结构以及所有其它进程的编号.每个进程的初始状态中有一个值为 0 或 1 的输入变量.网络链路可能发生失败而丢失消息.问题的求解目标是:所有进程都必须做出输出 0 或者 1 的一致判断;所有进程的判断必须符合下述三个条件:

- **一致性**　没有两个进程有不同的判断值.

- **有效性**

1° 如果所有进程以 0 开始,那么 0 是唯一可能的判断值.

2° 如果所有进程以 1 开始并且所有信息被递交,那么 1 是唯一可能的判断值.

- **终止性**　所有进程要在固定的 $r \geqslant 1$ 轮内做出判断.

假设消息丢失是由一些敌手(adversary)决定的.为了说明敌手的作用,引入**通信模式**(communication pattern)这一概念.一个通信模式 γ 是下述集合的任意一个子集:

$$\{(i,j,k):(i,j)\text{ 是图中的边}, k \geqslant 1\}.$$

如果对任一元素 $(i,j,k) \in \gamma, k \leqslant r$,则称 γ 是一个好的通信模式.如果 $(i,j,k) \in \gamma$,那么说明由 i 送往 j 的一个消息在第 k 轮递交成功.

敌手的作用是任意选择一个好通信模式,并为所有进程选择输入值.对任何特定的敌手而言,进程所做的随机选择集合将决定唯一的一次执行.也就是说,进程的随机选择在执行集合上引入了一个概率分布,用这种概率分布,可以表示所有进程做出一致选择的概率.为了突出敌手的作用,我们使用 P_γ^B 表示敌手 B 引入的概率函数.

带随机性的一致性判断的一致性可用概率形式表示如下:

- **一致性**　对每个敌手 B, P_γ^B[有些进程判断值为 0,有些进程判断值为 1] $\leqslant \epsilon$,其中 $0 \leqslant \epsilon \leqslant 1$.

联合攻击问题算法 RandomAttack 的形式化描述:

消息集由形如 (L,V,k) 的三元组构成,其中 L 是一个向量,每个向量元素对应一个进程索引,取值为 $[0,r]$ 中的一个整数;V 也是一个向量,每个向量元素对应一个进程索引,取值为集合 {0,1,undefined} 中的一个值;k 或者是 $[1,r]$ 范围内的整数,或者为 undefined.

```
states_i:
round_s ∈ N(N 为正整数集),initially 0
decision ∈ {unknown,0,1},initially unknown
key ∈ [1,r] ∪ undefined,initially undefined
for every j,1 ≤ j ≤ n,
val(j) ∈ {0,1,undefined};initially val(i) is i's initial value and
        val(j) = underfined for all j ≠ i
        level(j) ∈ [-1,r];initially level(i) = 0 and level(j) = -1 for all j ≠ i
rand_i:
if i = 1 and rounds = 0 then key: = random /* 第一个进程为所有进程随机选择
一个用于决定判断值的随机数 key */
msg_i
send (L,V,key) to all j,where L is the level vector and V is the val vector
trans_i:
```

```
rounds: = rounds + 1
let (L, V, k_j) be the message from j, for each j from which a message arrives
if some k_j ≠ undefined then key: = k_j;
for all j ≠ i do
    if some V_i'(j) ≠ undefined then val(j): = V_i'(j)
    if some L_i'(j) ≠ undefined then level(j): = max{L_i'(j)}
level(i): = 1 + min{level(j): j ≠ i}
if rounds = r then
    if key ≠ undefined and level(i) ≥ key and val(j) = 1 for all j then decision:
= 1
    else decision: = 0
```

可以证明,联合攻击问题算法 RandomAttack 解决联合攻击问题的概率为 $\epsilon = 1/r$,即对每个敌手 B,P_γ^B[有些进程判断值为 0,有些进程判断值为 1] $\leqslant 1/r$.

2. 有进程失败的分布式一致性问题

问题描述:一个连通的无向图 G 有 n 个节点 $1,2,\cdots,n$,节点上的每个进程都知道这个图的信息.每个进程有唯一的一个取自集合 V 中的输入值.假设通信链路完全可靠(即所有发出的消息都被正确递交).进程可能发生停止失败或者拜占庭失败,但是假定最多有 f 个进程失败.问题的目标是:进程输出一个取自集合 V 的判断值,并符合下述正确性条件:

在停止失败发生时:

- **一致性**　没有两个进程决定不同的值.
- **有效性**　如果所有进程开始的初始值都为 $v \in V$,那么 v 是唯一可能的判断值.
- **可终止性**　所有未发生失败的进程最终都做出判断.

在拜占庭失败发生时(称为拜占庭一致性问题):

- **一致性**　没有两个未发生失败的进程最终决定不同的值.
- **有效性**　如果所有未发生失败的进程的初始值为 v,那么 v 是唯一可能的判断值.
- **可终止性**　同上.

停止失败问题算法:

假定 v_0 表示输入集 V 中的缺省值,b 代表 V 中任何一个值需要的表示位数的上限.

算法一　FloodSet 算法

形式化描述:

消息集是 V 的子集.

$states_i$:

```
round_i ∈ N(N 为正整数集), initially 0
decision ∈ V ∪ {unknown}, initially unknown
W is subset of V, initially the sigleton set consisting of i's initial value

msg_i:
if rounds ≤ f then send W to all processes

trans_i:
rounds: = rounds + 1
let X_j be the message from j, for each j from which a message arrives
W: = W ∪ (⋃_j X_j)
if rounds = f + 1 then
    if |W| = 1 then decision: = v, where W = {v}
    else decision: = v_0
```

该算法的思想是:每个进程维护一个变量 W, W 包含 V 的一个子集.在进程 i 中, W 的初始值为 i 的输入值.在总共 $f+1$ 轮的每一轮中,每个进程都广播 W,并且将收到的所有集合中的元素加到 W 中. $f+1$ 轮之后,如果 W 只包含1个元素,则 i 的判断值为 W 中的元素,否则 i 的判断值为缺省值.

复杂度:FloodSet算法的时间复杂度为 $f+1$ 轮,通信复杂度为 $O((f+1)n^2)$,以位表示的通信位复杂度为 $O((f+1)n^3b)$.

算法二　指数信息收集(exponential information gathering)算法 EIGStop

首先定义带标记的 EIG 树 $T = T_{n,f}$:

从根(0)到叶($f+1$), T 有 $f+2$ 级.在级 $k, 0 \leq k \leq f$,每个结点有 $n-k$ 个儿子. T 中的每一个结点由一个进程索引串标记:根节点用空串 λ 标记.标记为 $i_1 i_2 \cdots i_k$ 的结点有 $n-k$ 个标记为 $i_1 i_2 \cdots i_k j$ 的儿子,其中 $j = \{1, 2, \ldots, n\} - \{i_1, i_2, \cdots, i_k\}$.从根节点开始的一条路径表示传播初始输入值的一条进程链.每个进程都有一棵 EIG 树.

算法描述:

对进程 i 的 EIG 树 T 上的每个结点标记 x,设置一个变量 val(x)记录 i 在 x 对应结点上的设置值.根结点的设置值 val(λ)为进程 i 的初始输入值.标识 $i_1 i_2 \cdots i_k$, $1 \leq k \leq f+1$ 的设置值 v 的含义是:在第 k 轮 i_k 告诉 i, i_{k-1} 在第 $k-1$ 轮告诉 i_k, $\cdots$, i_1 告诉 i_2, i_1 的初始值为 v,如果 $i_1 i_2 \cdots i_k$ 的设置值为 null,则说明通信链 $i_1 i_2 \cdots i_k$ 被切断了.

第1轮:

进程 i 首先将 val(λ)广播给所有的进程(包括 i 自己).然后记录接受到的信息:

(1) 如果值为 $v \in V$ 的消息从 j 到达了 i,则 val(j): = v.

(2) 如果没有从 j 收到消息,则 val(j): = null.

第 k 轮($2 \leqslant k \leqslant f+1$):

进程 i 广播所有的对(x, v),其中 x 为 T 的第 $k-1$ 级标记且不包含索引 i, v = val(x), $v \in V$. 然后记录接受到的信息:

(1) 如果 x_j 是 T 的第 k 级结点标识(x 是一个进程索引串, j 是单个索引),从 j 收到了一个消息(x, v),则 val(x_j): = v.

(2) 如果 x_j 是 T 的第 k 级结点标识,但没有从 j 收到消息,则 val(x_j): = null. 在第 $f+1$ 轮末尾,进程 i 使用如下判断规则: W 为所有非 null 设置值的集合,如果 W 只包含一个元素,则 i 的判断值即为该元素,否则 i 的判断值为 v_0.

复杂度:EIGStop 算法的时间复杂度为 $f+1$,消息复杂度是 $O((f+1)n^2)$,以位表示的消息复杂度是 $O(n^{(f+1)}b)$.

算法三　拜占庭一致性问题的 EIG 算法 EIGByz

所有进程像 EIPStop 算法中那样传播 $f+1$ 轮消息,但做如下修改:

(1) 如果进程 i 从进程 j 收到了一条消息,但该消息不符合正确的形式(如完全是垃圾, j 中的同一个结点有重复值),则说明 j 发生了拜占庭失败,进程 i 扔掉该消息,并当作没有从 j 收到消息.

(2) 在第 $f+1$ 轮结尾,进程 i 将所有的 null 值用缺省值 v_0 代替.

(3) 判断规则改为: i 为经过第 2 步工作的 EIG 树中的每个结点设置另外一个设置值 newval, 并从叶结点开始, 依次向上进行如下赋值: 对所有的叶结点, x: newval(x): = val(x); 对标记为 x 的非叶结点, newval(x) 为其所有儿子中严格多数(strict majority) 儿子所具有的 newval. 即如果标记形式为 x_j 的所有结点中,严格多数结点的 newval 为 v, 则 newval(x): = v, $v \in V$. 如果严格多数不存在, 则 newval(x): = v_0. 进程 i 的最终判断值为 newval(λ).

复杂度:EIGByz 算法的复杂度同 EIGStop. 但该算法的前提是 $n > 3f$, 即进程树大于最大可能失败数的 3 倍.

3. k 一致性(k-agreement) 算法 FloodMin:

问题描述:

k 一致性问题与发生停止失败的分布式一致性问题基本相同,只是正确性条件不同:

- 一致性:存在 V 的子集 W, $|W| = k$,所有的判断值均在 W 中.
- 有效性:任何进程的判断值是某些进程的初始值.
- 可终止性:所有未失败的进程最终要做出判断.

算法的形式化描述:

消息集是 V.

states$_i$:

round$_s$ ∈ N(N 为正整数集), initially 0

decision ∈ V ∪ {unknown}, initially unknown

min-val ∈ V, initially i's initial value

msg_i:
if rounds ≤ [f/k], then send min-val to all processes

$trans_i$:
rounds: = rounds + 1
let m_j be the message from j, for each j from which a message arrives
min-val: = min({min-val ∪ {m_j: j ≠ i})
if rounds = [f/k] + 1 then decision: = min-val

算法的主要思想是:每个进程维护一个变量 min-val,其初值为 i 的初始值.在 $[f/k]+1$ 轮中的每一轮,所有进程都广播自己的 min-val,然后每个进程将 min-val 设置为老的 min-val 和所有接受到的消息的最小值.进程的判断值为最后的 min-val.

复杂度:FloodMin 算法的时间复杂度为 $[f/k]+1$ 轮,通信复杂度为 $([f/k]+1)n^2$,用位表示的通信复杂度为 $([f/k]+1)n^2b$.

参 考 文 献

1 Richard M K. Parallel algorithms for shared-memory Machines. In: Handbook of Theoretical Computer Science. New York: Elsevier Science Publishers B V, 1990. 870 ~ 941.
2 Akl S G. The design and analysis of parallel algorithms. Englewood Cliffs: Prentice-Hall, 1989.
3 Jaja J. An introduction to parallel algorithms. Reading, MA: Addison-Wesley. 1992.
4 Kumar V Grama A, Gupta A, Karypis G. Introduction to parallel computing: design and analysis of algorithms. The Benjamin/Cummings Publishing Company, 1994.
5 Valiant L G. A bridging model for parallel computation. Communications of the Association for Computing Machinery, 1990, 33(8): 103 ~ 111.
6 David Culler, Richard Karp, David Patterson, etc. LogP: towards a realistic model of parallel Computation. Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, May 1993, San Diego, CA.
7 Ranade A G. How to emulate shared memory. In proceedings of the 28th IEEE annual Symposium on Foundations of Computer Science, 1987, 185 ~ 194
8 Karp R M, Luby M an F Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. In proceedings of the 24th Annual ACM Symposium of the Theory of Computing. 1992, May 318 ~ 326
9 Nancy A Lynch. Distributed algorithms. Morgan Kaufmann Publishers Inc, 1996.
10 Howard Jay Siegel. Inter-connection networks for large-scale parallel processing (theory

and Case Studies). New York: McGraw-Hill Publishing Company, 1990.
11 Leslie Lamport, Nancy Lynch. Distributed computing: models and methods. In: Jan Van Leeuwen editor. Formal Models and semantics, volume B of handbook of theoretical computer science, New York: Elsevier/MIT Press, 1990. 1157 ~ 1199.
12 Utpal Banerjee. Dependence Analysis. Boston: Kluwer Academic Publishers, 1997.

·计算机数学卷·

第 20 篇

计算几何

编　者　孙家广
审校者　华宣积

目　录

引　言

现代科技的发展，特别是汽车、航空等现代工业的发展，需要用计算机技术设计表示复杂的物体外形曲面，这类曲面难以用传统的几何学来研究．传统的解析几何主要研究二次曲线、曲面，微分几何虽然也讨论一些具体的曲线、曲面，但更多地研究正则曲线、曲面的内在的几何性质．构造物体复杂外形的研究已发展成为一门称为计算机辅助几何设计(CAGD)的新兴交叉学科．

法国雷诺(Renault)汽车公司的贝齐尔(Bezier)和法国雪铁龙(Citroen)汽车公司的德卡斯特里奥(de Casteljau)，在20世纪60年代初期由于汽车外形设计的需要分别独立研究出用控制多边形和控制网格定义曲线和曲面的方法，并提出了一种几何直观性很强的递推算法．1964年美国麻省理工学院(MIT)的孔斯(Coons)用四条边界定义一块曲面，提出了一种重要的插值方法．1974年，美国通用汽车公司的戈登(Gordon)和里森费尔德(Riesenfeld)将B样条理论用于形状描述，提出了B样条曲线与曲面．20世纪80年代后期，美国的皮格尔(Piegl)和蒂勒(Tiller)将有理B样条发展成非均匀有理B样条(NURBS)方法，并已成为当前自由曲线和曲面描述中最为流行的技术。用NURBS可统一表示初等解析曲线、曲面以及有理与非有理贝齐尔曲线、曲面和非有理B样条曲线、曲面等．国内如苏步青等人在20世纪70年代初期由于造船工业的需要也开始了这方面的研究，我国学者在参数曲线的仿射不变性、曲线曲面的拼接条件等领域也有出色的研究成果．

1　曲线、曲面基础

1.1　曲线、曲面的表示

1.1.1　曲线、曲面的显式、隐式和参数表示

在几何设计或计算机图形学中，大部分曲线、曲面主要表示为两种形式：一种是作为实多项式的零点集的隐式定义，另一种是有理参数形式定义，其中用有理参数形式定义的曲线、曲面更为常见．

设$f(x,y)$是二元实系数多项式，在$\mathbf{R}^2$中隐式方程

$$f(x,y)=0 \tag{1-1}$$

给出的图形称为**实代数曲线**，$f(x,y)$的次数称为该代数曲线的次数．常见的实代数曲线有直线、圆、椭圆、双曲线、抛物线等．对于三元n次实多项式$f(x,y,z)$，相应的隐式方程

$$f(x,y,z)=0 \tag{1-2}$$

定义的曲面称为 **n 次实代数曲面**. 两个代数曲面的交:

$$\begin{cases} f(x,y,z) = 0 \\ g(x,y,z) = 0 \end{cases} \tag{1-3}$$

是一条空间曲线, 称为**空间代数曲线**. 通常所指的隐式表示是指曲线(1-1)、曲面(1-2)的表示形式. 若(1-1)可表示成

$$y = f(x), \tag{1-4}$$

(1-2)式可表示成

$$z = f(x,y), \tag{1-5}$$

则称(1-4)(或(1-5))式为曲线(或曲面)的显式表示.

若(1-1)、(1-2)式分别可表示成

$$\begin{cases} x = x(t), \\ y = y(t), \end{cases} \qquad \begin{cases} x = x(u,v), \\ y = y(u,v), \\ z = z(u,v), \end{cases}$$

则称曲线(1-1)、曲面(1-2)可表示成参数形式.

在曲线、曲面的表示上, 参数表示与隐式表示各有所长. 两者的区别:

(1) 参数形式有更大的自由度来控制曲线、曲面的形状.

(2) 对非参数方程表示的曲线、曲面进行变换, 必须对曲线、曲面上的每个型值点进行几何变换; 而对参数表示的曲线、曲面可对其参数方程直接进行几何变换(如平移、比例、旋转), 从而节省计算工作量.

(3) 便于处理斜率为无限大的问题, 不会因此而中断计算.

(4) 参数方程中, 代数、几何相关和无关的变量是完全分离的, 而且对变量个数不限, 从而便于用户把低维空间中的曲线、曲面扩展到高维空间去. 这种变量分离的特点使我们可以用数学公式去处理几何分量, 如调配函数就具有此特点.

(5) 规格化的参数变量 $t \in [0,1]$, 使其相应的几何分量是有界的, 而不必用另外的参数去定义其边界. 参数表示易于产生曲线上的有序点列和曲面上的有序网格点, 而隐式表示则无此特点.

(6) 易于用矢量和矩阵表示几何分量, 简化了计算.

(7) 用参数形式设计或表示形状更直观, 许多参数表示的基底如伯恩斯坦(Bernstein)基函数和 B 样条函数, 具有明显的几何意义, 由此得到的算法更直观稳定.

同时隐式表示也有长处:

(8) 用隐式表示更容易表示曲面的等值线. 另外在曲线或曲面的求交中, 通过隐式方程容易判断某点是否落在曲线或曲面上, 而参数表示则无此优点.

(9) 参数形式有时需要处理与几何形状无关的参数异常点, 例如单位球面的极点, 但事实上, 极点与球面上其它点并无区别.

(10) 曲面上参数表示并不能描述曲面间的相互位置, 找到它们之间的整体关系很困难, 而隐式表示则较为方便.

1.1.2 参数曲线的代数形式和几何形式

一条三次参数曲线的代数形式是

$$\begin{cases} x(t) = a_{3x}t^3 + a_{2x}t^2 + a_{1x}t + a_{0x}, \\ y(t) = a_{3y}t^3 + a_{2y}t^2 + a_{1y}t + a_{0y}, \qquad t \in [0,1]. \\ z(t) = a_{3z}t^3 + a_{2z}t^2 + a_{1z}t + a_{0z}, \end{cases}$$

从 a_{3x} 到 a_{0z} 这 12 个系数，确定了一条参数曲线的形状和位置.上述代数式写成矢量形式是

$$\boldsymbol{p}(t) = \boldsymbol{a}_3 t^3 + \boldsymbol{a}_2 t^2 + \boldsymbol{a}_1 t + \boldsymbol{a}_0, \quad t \in [0,1], \tag{1-6}$$

$\boldsymbol{p}(t)$ 表示曲线上任意一点的位置矢量，其分量对应于直角坐标系中该点的坐标，$\boldsymbol{a}_0, \boldsymbol{a}_1, \boldsymbol{a}_2$ 和 $\boldsymbol{a}_3$ 是代数系数矢量.

对于空间曲线，用于描述曲线的可供选择的条件有：端点坐标、切矢量、曲率和挠率等.例如对于(1-6)式，用两个端点 $\boldsymbol{p}(0)$、$\boldsymbol{p}(1)$ 以及对应的切矢量 $\boldsymbol{p}'(0) = \mathrm{d}\boldsymbol{p}(0)/\mathrm{d}t$，$\boldsymbol{p}'(1) = \mathrm{d}\boldsymbol{p}(1)/\mathrm{d}t$，记 $\boldsymbol{p}_0 = \boldsymbol{p}(0)$，$\boldsymbol{p}_1 = \boldsymbol{p}(1)$，$\boldsymbol{p}'_0 = \boldsymbol{p}'(0)$，$\boldsymbol{p}'_1 = \boldsymbol{p}'(1)$，则可得到下列形式：

$$\begin{aligned} \boldsymbol{p}(t) = {} & (2t^3 - 3t^2 + 1)\boldsymbol{p}_0 + (-2t^3 + 3t^2)\boldsymbol{p}_1 + \\ & (t^3 - 2t^2 + t)\boldsymbol{p}'_0 + (t^3 - t^2)\boldsymbol{p}'_1, \quad t \in [0,1], \end{aligned} \tag{1-7}$$

令

$$F_0 = 2t^3 - 3t^2 + 1, \quad F_1 = -2t^3 + 3t^2,$$
$$G_0 = t^3 - 2t^2 + t, \quad G_1 = t^3 - t^2;$$

重写(1-7)式有

$$\boldsymbol{p}(t) = F_0\boldsymbol{p}_0 + F_1\boldsymbol{p}_1 + G_0\boldsymbol{p}'_0 + G_1\boldsymbol{p}'_1. \tag{1-8}$$

(1-8)式是**参数曲线的几何形式**，$\boldsymbol{p}_0$、$\boldsymbol{p}_1$、$\boldsymbol{p}'_0$、$\boldsymbol{p}'_1$ 为其几何系数，(F_0, F_1, G_0, G_1) 为三次**埃尔米特(Hermite)调配函数**.

(1-6)式可写成

$$\boldsymbol{p} = \boldsymbol{TA}, \tag{1-9}$$

其中 $\boldsymbol{T} = [t^3\ t^2\ t\ 1]$，$\boldsymbol{A} = [\boldsymbol{a}_3\ \boldsymbol{a}_2\ \boldsymbol{a}_1\ \boldsymbol{1}]^{\mathrm{T}}$.(1-8)式可写成

$$\boldsymbol{p} = \boldsymbol{FB}, \tag{1-10}$$

其中 $\boldsymbol{F} = (F_0, F_1, G_0, G_1)$，$\boldsymbol{B} = [\boldsymbol{p}_0\ \boldsymbol{p}_1\ \boldsymbol{p}'_0\ \boldsymbol{p}'_1]^{\mathrm{T}}$.$\boldsymbol{A}$ 是代数系数矩阵，$\boldsymbol{B}$ 是几何系数矩阵或边界条件矩阵.用矩阵运算可推导出代数形式和几何形式之间的关系.

因为 $\boldsymbol{F} = (2t^3 - 3t^2 + 1, -2t^3 + 3t^2, t^3 - 2t^2 + t, t^3 - t^2)$，也可将此式写成：

$$\boldsymbol{F} = [t^3\ t^2\ t\ 1]\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \boldsymbol{TM}, \quad \boldsymbol{M}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix},$$

则(1-10)式可重写成：$\boldsymbol{p} = \boldsymbol{TMB}$，则有

$$\boldsymbol{A} = \boldsymbol{MB}, \quad \boldsymbol{B} = \boldsymbol{M}^{-1}\boldsymbol{A}.$$

上面两式反映了参数曲线代数形式和几何形式之间的变换关系.(1-10)式是

由端点及其切矢量定义的三次参数曲线,也称为**埃尔米特曲线**或**弗格森(Ferguson)曲线**.

1.2 数据点列的参数化

确定一条插值于 $n+1$ 个点 $p_i(i=0,1,\cdots,n)$ 的参数插值曲线,必须先给数据点 p_i 赋于相应的参数值 u_i,使其形成一个严格递增的序列 $\Delta u: u_0 < u_1 < \cdots < u_n$,其中每个参数值称为节点(knot)或分割点.这一过程称为关于 u 的一个参数分割.由此决定了曲线上点与参数域内相应点之间的对应关系.对一组有序数据点决定一个参数序列分割称之为对这些数据实行参数化(parameterization).同一组数据,即便采用同样的插值方法,若数据点的参数化不同,将可获得不同的插值曲线.常用的参数化方法如下:

1. 均匀参数化(又称等距参数化)法

使每个节点区间长度(用向前差分表示)$\Delta_i = u_{i+1} - u_i =$ 正常数,$i=0,1,\cdots,n-1$,即节点在参数轴上呈等距分布.为处理方便起见,常取成整数序列

$$u_i = i,\quad i = 0,1,\cdots,n.$$

这种参数化法仅适合于数据点多边形各边(或称弦)接近相等的场合.

2. 累加弦长参数化(或简称弦长参数化)法

$$\begin{cases} u_0 = 0, \\ u_i = u_{i-1} + |\Delta p_{i-1}|, \end{cases} \qquad i = 1,2,\cdots,n,$$

其中 Δp_k 为向前差分矢量,$\Delta p_k = p_{k+1} - p_k$ 即弦线矢量.这种参数化法反映了数据点按弦长的分布情况,弦长参数化法生成的插值曲线在某种程度上可看作近似的弧长参数化,切矢模长较接近单位长.

3. 向心参数化法

$$\begin{cases} u_0 = 0, \\ u_i = u_{i-1} + |\Delta p_{i-1}|^{1/2}, \end{cases} \qquad i = 1,2,\cdots,n.$$

与累加弦长参数化法的差别在于弦长平方根的累加,故又称为平方根法.

4. 修正弦长参数化法

$$\begin{cases} u_0 = 0, \\ u_i = u_{i-1} + k_i |\Delta p_{i-1}|, \end{cases} \qquad i = 1,2,\cdots,n.$$

其中 $k_i = 1 + \dfrac{3}{2}\left(\dfrac{|\Delta p_{i-2}|\theta_{i-1}}{|\Delta p_{i-2}| + |\Delta p_{i-1}|} + \dfrac{|\Delta p_i|\theta_i}{|\Delta p_{i-1}| + |\Delta p_i|}\right)$,

$$\theta_i = \min\left(\pi - \angle p_{i-1}p_ip_{i+1}, \frac{\pi}{2}\right),\quad |\Delta p_{-1}| = |\Delta p_n| = 0.$$

可见这里采用了修正弦长,修正系数 $k_i \geqslant 1$.与前后邻弦长 $|\Delta p_{i-2}|$ 及 $|\Delta p_i|$ 相比,若弦长 $|\Delta p_{i-1}|$ 越小,且与前、后邻弦线夹角的外角 θ_{i-1}、θ_i(不超过 $\pi/2$ 时)越大,则修正系数 k_i 就越大,因而修正弦长即参数区间 $\Delta_{i-1} = k_i|\Delta p_{i-1}|$ 也就越大.当该曲线段绝对曲率偏大时,这一方法对实际弦长比弧长偏短的情况起到了修正作用.修正弦长就较接近于实际弧长.

图 1-1

图 1-1 显示了用同一组数据点，采用不同参数化法生成不同的参数多项式插值曲线（虚线、实线、点划线与双点划线分别表示用均匀参数化、弦长参数化、向心参数化与修正弦长参数化法）.

1.3　调配函数

构造参数样条的方法是：将曲线关于某一种基底进行分解，这些基函数称为调配函数．常见的调配函数有埃尔米特基函数、伯恩斯坦基函数、B 样条基函数等．此外，还有拉格朗日（Lagrange）基和幂基等．

1.3.1　埃尔米特基函数

在 1.1.2 节中，(1-8) 式中的 F_i、$G_i(i=0,1)$ 就是三次埃尔米特调配函数：

$$F_0(t)=1-3t^2+2t^3,$$

$$F_1(t)=3t^2-2t^3=1-F_0(t),$$

$$G_0(t)=t(1-t)^2,$$

$$G_1(t)=-t^2(1-t)=-G_0\frac{t}{1-t},$$

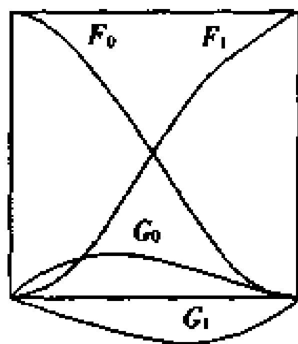


图 1-2

它们具有如下形式：

$$F_i(j)=G'_j(j)=\delta_{ij}=\begin{cases}1, & i=j,\\0, & i\neq j,\end{cases}\quad i,j=0,1.$$

$$F'_i(j)=G_i(j)=0.$$

如(1-8)式所示，F_i、G_i 都具有明确的几何意义（如图 1-2），这在曲线、曲面插值使用中方便直观.

1.3.2　伯恩斯坦基函数

伯恩斯坦基函数的形式如下

$$B_{i,n}(t)=C_n^i t^i(1-t)^{n-i},\quad i=0,1,\cdots,n,0\leqslant t\leqslant 1.$$

图 1-3 是五次伯恩斯坦基函数的图形．伯恩斯坦基函数具有如下性质：

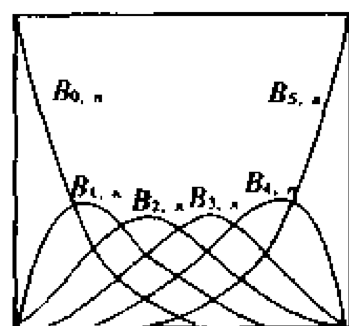


图 1-3

(1) $B_{i,n}(t)\begin{cases} = 0, & t = 0,1, \\ > 0, & t \in (0,1), \end{cases} i = 1,2,\cdots,n-1,$

并且

$$\begin{cases} B_{0,n}(0) = B_{n,n}(1) = 1, \\ B_{0,n}(1) = B_{n,n}(0) = 0, \\ 0 < B_{0,n}(t), B_{n,n}(t) < 1, \quad t \in (0,1). \end{cases}$$

(2) 权性　$\sum_{i=0}^{n} B_{i,n}(t) = 1, t \in (0,1).$

(3) 对称性　$B_{i,n}(t) = B_{n-i,n}(1-t),\ i = 0,1,\cdots,n.$

(4) 递推性　$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t),\quad i = 0,1,\cdots,n.$

$$B_{i,n}(t) = \left(1 - \frac{i}{n+1}\right)B_{i,n+1}(t) + \frac{i+1}{n+1}B_{i+1,n+1}(t),$$
$i = 0,1,\cdots,n.$

(5) 最大值　$B_{i,n}(t)$ 在 $t = \frac{i}{n}$ 处达到最大值.

(6) 分割　$B_{i,n}(ct) = \sum_{k=i}^{n} B_{k,n}(t)B_{i,k}(c).$

(7) 积分　$\int_0^1 B_{j,n}(t)\mathrm{d}t = \frac{1}{n+1}.$

(8) 与幂基的关系
$$\begin{cases} t^i = \sum_{j=i}^{n} \frac{C_j^i}{C_n^i} B_{j,n}(t), \\ B_{j,n}(t) = \sum_{i=j}^{n} (-1)^{i+j} C_n^i C_i^j t^i. \end{cases}$$

1.3.3　B样条基函数

B样条基函数是样条函数空间中具有最小支承的一组基，也称为基本样条(basic spline)，B样条基函数有多种定义形式，例如有积分、差分、卷积、截断幂、递推形式等，理论上较多采用截断幂函数的差商定义，但实际工程计算则更多采用de Boor-Cox递推公式定义B样条基函数. 这样不仅便于计算分析基函数的性质，而且也简明直观，具有明显的几何特征.

定义1　设 $U = (u_0, u_1, \cdots, u_m)$，$\{u_i\}$ 是单调递增的实数序列，即 $u_i \leqslant u_{i+1}(i = 0,1,\cdots,m-1)$，$u_i$ 称为节点(knot)，U 称为节点矢量. 第 i 段 p 次(或 $p+1$ 阶)B样条基函数记为 $N_{i,p}(u)$，定义如下：

$$N_{i,0}(u) = \begin{cases} 1, & u_i \leqslant u < u_{i+1}, \\ 0, & \text{否则}, \end{cases} \tag{1-11}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u),$$

并且规定：$\frac{0}{0} = 0.$

B样条基函数的递归定义如图1-4所示.

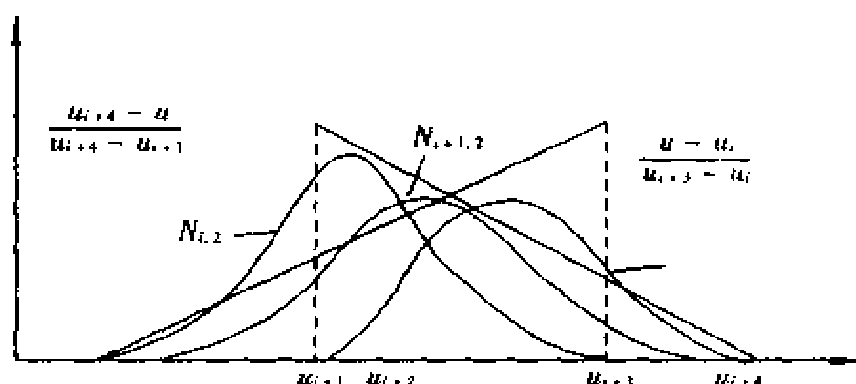


图1-4

B样条基函数的性质:

(1) 局部支撑性和非负性 $N_{i,p}(u)\begin{cases}\geqslant 0, & u\in[u_i,u_{i+p+1}],\\ =0, & \text{其它}.\end{cases}$

(2) 任意给定节点区间$[u_j,u_{j+1})$,最多只有 $p+1$ 个 $N_{i,p}$ 非零;$N_{i-p,p}(u),\cdots,N_{i,p}(u)$ 非零.

(3) 规范性 对于任一 $u\in[u_i,u_{i+1})$,$\sum\limits_j N_{j,p}(u)=\sum\limits_{j=i+1-p}^{i}N_{j,p}(u)=1$.

(4) 除 $p=0$ 外,$N_{i,p}(u)$ 存在一个最大值.

(5) 在节点区间内部,$N_{i,p}(u)$ 任意阶可导,而在节点处 $N_{i,p}(u)$ 是 $p-k$ 次可导,其中 k 是该节点的重数.因此升阶可增加连续性;降低节点重数,也可增加连续性.

(6) 由递推公式,$N_{i,p}(u)$ 的 k 阶导函数可表示为

$$N_{i,p}^{(k)}(u)=(p-1)\left[\frac{N_{i,p-1}^{(k-1)}}{u_{i+p-1}-u_i}-\frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p}-u_{i+1}}\right].$$

对于以下形式的非周期节点矢量:

$$U=\{\underbrace{a,\cdots,a}_{p+1},u_{p+1},\cdots,u_{m-p-1},\underbrace{b,\cdots,b}_{p+1}\},\tag{1-12}$$

其B样条基函数有下列性质:

(7) 设节点个数为 $m+1$,则有 $n+1$ 个基函数,这里 $n=m-p-1$,$N_{0,p}(a)=1$,和 $N_{n,p}(b)=1$.

(8) 若节点矢量是如下形式:

$$U=\{\underbrace{0,0,\cdots,0}_{p+1},\underbrace{1,1,\cdots,1}_{p+1}\}$$

则 $N_{i,p}(u)$ 变成了 p 次伯恩斯坦基函数.由此可见,伯恩斯坦基函数是B样条基函数的特例.

1.3.4　有理基函数

1. 有理伯恩斯坦基函数

$$R_{i,n}(u) = \frac{w_i B_{i,n}(u)}{\sum_{j=0}^{n} B_{j,n}(u) w_j}, \quad u \in [0,1], \quad w_i > 0. \tag{1-13}$$

它具有与伯恩斯坦基函数 $B_{i,n}(u)$ 相似的性质：

(1) $R_{i,n}(u) \geqslant 0$.

(2) 单位分割　$\sum_i R_{i,n}(u) = 1$.

(3) 端点插值 $R_{0,n}(0) = R_{n,n}(1) = 1$.

(4) $R_{i,n}(u)$ 存在最大值.

(5) 对任意 $i, w_i = 1$ 则 $R_{i,n}(u) = B_{i,n}(u)$.

2. 有理 B 样条基函数

对应于 B 样条基函数,定义有理基函数:对于任意 $i, w_i > 0$,

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^{n} N_{j,p}(u) w_j}, \qquad u \in [0,1]. \tag{1-14}$$

相应于定义(1-11),有理基函数具有下列性质：

(1) 非负性　对于任一 i、p 以及 $u \in [0,1]$, $R_{i,p}(u) \geqslant 0$.

(2) 单位分割　对 $\forall u \in [0,1]$, $\sum_{i=0}^{n} R_{i,p}(u) = 1$.

(3) $R_{0,p}(0) = R_{n,p}(1) = 1$.

(4) 对 $p > 0$,所有 $R_{i,p}(u)$ 在[0,1]上存在一最大值.

(5) 局部支承性　$R_{i,p}(u) = 0$ 对于 $u \bar{\in} [u_i, u_{i+p+1})$,并且对于任一节点区间,最多只有 $p+1$ 个 $R_{i,p}(u)$ 非 0(一般地,$[u_i, u_{i+1})$ 中只有 $R_{i-p,p}(u) \cdots, R_{i,p}(u)$ 非零).

(6) 在节点区间内部,$R_{i,p}(u)$ 是任意阶可微,在节点处,$R_{i,p}(u)$ 是 p-k 次连续可微的,这里 k 是该节点的重数.

(7) 对于所有 i,如 $w_i = 1$,则 $R_{i,p}(u) = N_{i,p}(u)$,即 $N_{i,p}(u)$ 是 $R_{i,p}(u)$ 的特例.

1.3.5　张量积 B 样条基函数

设 $N_{i,p}(u)$、$N_{i,q}(v)$ 是两个如(1-11)式所定义的单变元 B 样条基函数,相应的节点矢量分别为

$$U = (\underbrace{0,0,\cdots,0}_{p+1}, u_{p+1}, \cdots, u_{r-p-1}, \underbrace{1,\cdots,1}_{p+1}),$$

$$V = (\underbrace{0,\cdots,0}_{q+1}, u_{q+1}, \cdots, u_{s-q-1}, \underbrace{1,\cdots,1}_{q+1}).$$

$\boldsymbol{U}$ 有 $r+p+1$ 个节点，$\boldsymbol{V}$ 有 $s+q+1$ 个节点，则两基函数的乘积 $N_{i,p}(u)N_{i,q}(v)$ 称为**张量积的 B 样条基函数**. 它具有如下性质：

(1) 非负性　$N_{i,p}(u)N_{i,q}(v) \geqslant 0$ 对所有 i、j、p、q、u、v 都成立.

(2) 单位分割　设 $n = r-p-1, m = s-q-1$，对所有 $(u,v) \in [0,1] \times [0,1]$，

$$\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{i,q}(v) = 1.$$

(3) 若 $n = p, m = q, \boldsymbol{U} = (0,\cdots,0,1,\cdots,1), \boldsymbol{V} = (0,\cdots,0,1,\cdots,1)$，则对所有 i、j，$N_{i,p}(u)N_{j,q}(v) = B_{i,n}(u)B_{j,m}(v)$，即 B 样条基函数的乘积化为伯恩斯坦基函数的乘积；

(4) 若 $(u,v) \in [u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$，则 $N_{i,p}(u)N_{j,q}(v) = 0$.

(5) 对任一矩形域 $[u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$，最多只有 $(p+1)(q+1)$ 个基函数非零，从而对 $i_0 - p < i \leqslant i_0, j_0 - q \leqslant j \leqslant j_0$，$N_{i,p}(u)N_{j,q}(v) \neq 0$.

(6) 若 p、$q > 0$，则 $N_{i,p}(u)N_{j,q}(v)$ 存在最大值.

基函数 $N_{j,p}(u)N_{i,q}(v)$ 是双变量多项式，它的所有偏导数都存在，对于一节点 u（相应地，节点 v），它在 u（相应地，v）方向是 $p-k$（相应地，$q-k$）次可微，这里 k 是节点的重数.

1.4　曲线、曲面间的几何连续性

复杂的几何外形设计对曲线、曲面的操作通常需要三个步骤：曲线、曲面的构造，曲线、曲面间的拼接以及曲线、曲面的修改. 因此，曲线、曲面间的拼接问题是曲线、曲面研究的重要内容.

1.4.1　参数曲线间的连续性条件

曲线间的连续性条件有三种形式：参数连续（parametric continuity），几何连续（geometric continuity）和弗朗内标架连续（Frenet frame continuity）.

1. 参数连续性

参数连续就是通常意义下的连续，与参数选择有关.

定义2　参数曲线 $\boldsymbol{C}(t)$ 在 t_0 处是 **m 阶参数连续**（C^m 的），当且仅当 $\boldsymbol{C}(t)$ 的每一个分量函数在 t_0 处是 C^m 连续的，即

$$\lim_{t\to t_0^+} \boldsymbol{C}^{(i)}(t) = \lim_{t\to t_0^-} \boldsymbol{C}^{(i)}(t), \quad i = 0,\cdots,m,$$

其中 $\boldsymbol{C}^{(i)}(t)$ 表示向量函数的 i 阶导数.

若对所有的 $t \in [a,b]$，曲线 $\boldsymbol{C}(t)$ 都满足参数连续条件，那么 $\boldsymbol{C}(t)(a \leqslant t \leqslant b)$ 是一段关于 t 的参数光滑曲线，即 C^m 曲线. 在构造组合曲线时，每一段曲线通常是按照局部参数定义的 C^m 曲线段，因此要得到满足参数连续性条件的组合曲线，实质是要找一个合适的整体参数，使得曲线关于它是整体光滑的.

2. 几何连续性

几何连续性是参数连续性的推广，其实质是局部再参数化的存在性.

定义3　对于正则曲线段 $C_1(u)$、$C_2(t)$，$u\in[a,b]$，$t\in[c,d]$，两曲线在 $t=t_0$ 和 $u=u_0$ 处相接，若存在一个参数变换(C^m 微分同胚)$\varphi:[a,b]\to[c,d]$，使得 $C_1(u)$、$C_2(\varphi^{-1}(u))$ 在相接处是 C^m 的，则称它们的连接是 $G^m(GC^m)$ 连续的.

几何连续性有多种等价定义，也可称为 G^m 连续性条件.

由定义知，对于 $C_1(u)$，取变换 $u=u(t)$，且在公共点 P 处有参数 $u_0=u(t_0)$，则 $C_1(u)$ 与 $C_2(t)$ 在 P 点处是 C^m 连续的条件为(不妨设 $C_1(u)$ 在 $C_2(t)$ 的左侧)

$$
\begin{aligned}
&C_1(u_0-0)=C_2(t_0+0),\\
&C'_1(u_0-0)=\frac{\mathrm{d}u}{\mathrm{d}t}C'_2(t_0+0),\\
&C''_1(u_0-0)=\frac{\mathrm{d}^2u}{\mathrm{d}t^2}C'_2(t_0+0)+\left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)^2C''_2(t_0+0).
\end{aligned}
$$

令 $\beta_1=\dfrac{\mathrm{d}u}{\mathrm{d}t},\beta_2=\dfrac{\mathrm{d}^2u}{\mathrm{d}t^2},\cdots,\beta_n=\dfrac{\mathrm{d}^nu}{\mathrm{d}t^n}$，则有

$$
\begin{bmatrix} C_1\\ C'_1\\ C_1^{(2)}\\ \vdots\\ C_1^{(n)}\end{bmatrix}=\begin{bmatrix}1&&&&\\0&\beta_1&&&\\0&\beta_2&\beta_1^2&&\\ \vdots&\vdots&\vdots&\ddots&\\0&\beta_n&\cdots&\cdots&\beta_1^n\end{bmatrix}\begin{bmatrix} C_2\\ C'_2\\ C_2^{(2)}\\ \vdots\\ C_2^{(n)}\end{bmatrix},\tag{1-15}
$$

其中 $\beta_1>0$. 上述一组关系称为**贝塔(Beta)约束**，右端的方阵称为**关联矩阵**，它是一个下三角阵. 由此可得

条件一　两曲线段在拼接处具有 G^n 连续性的充要条件是，存在一组实数 β_i，$i=1,2,\cdots,n$，其中 $\beta_1>0$，使两曲线段在拼接处的两侧(左右)导矢满足贝塔约束条件(1-15).

利用贝塔约束，通过改变 β 值可以控制相邻两曲线段之一在拼接处附近的曲线形状，所以 β 值也称为**形状参数**. 对应两曲线间的 G^m 连续性，存在 m 个形状参数，这些形状参数为曲线形状控制提供了额外的自由度. 另外，还可以由此构造贝塔样条曲线.

条件二　两曲线段在公共连接处具有 G^m 的充要条件是，两曲线段相应的弧长参数化在拼接处具有 C^m 连续性.

它表明，在弧长参数化下，几何连续性与参数连续性是一致的. 条件二中曲线间关于弧长参数是 C^n 连续的，与微分几何中的 n 阶切触是一致的，而且利用拼接处的泰勒(Taylor)展开，由 n 阶切触定义可推得条件一中的贝塔约束(1-15). 因此可得下列等价条件：

条件三　曲线间在拼接处具有 G^n 连续或是 G^n 的充要条件是，曲线在连接处具有 n 阶切触.

3. 弗朗内标架连续性

弗朗内标架连续是几何连续的一种形式，它是基于高维空间中广义曲率概念及弗朗内标架的连续性来定义的，也是一种几何不变量.

设 $\boldsymbol{p}(s)$ 是 d 维空间的一条以弧长 s 为参数的曲线，则它的弗朗内标架（正交矢量）$\boldsymbol{v}_1(s),\boldsymbol{v}_2(s),\cdots,\boldsymbol{v}_d(s)$ 及其高阶曲率可按如下递推地定义：

$$\begin{cases} \boldsymbol{v}_1(s) = \boldsymbol{p}'(s), \\ K_0(s) = 0, \\ \boldsymbol{v}_{i+1}(s) = \dfrac{\boldsymbol{v}'_i(s) + K_{i-1}(s)\cdot \boldsymbol{v}_{i-1}(s)}{K_i(s)}. \end{cases} \tag{1-16}$$

此处 $K_i(s)$ 的选取应使 $\boldsymbol{v}_{i+1}(s)$ 具有单位长度. 在三维空间中，$K_1(s)$ 与 $K_2(s)$ 分别是曲线的曲率与挠率或二阶曲率. 各阶曲率都是曲线的几何不变量. 弗朗内标架连续性定义如下：

一条曲线是 n 阶弗朗内标架连续的（记为 F^n），当且仅当它的弧长参数化的弗朗内标架矢量和广义曲率是连续的.

弗朗内标架连续性和前述几何连续性对于一阶与二阶来说，两者是一致的. 当 $n>2$ 时，存在 F^n 连续的曲线，却非正则 C^n 连续的. 应该注意，d 维空间所有 F^d 连续的曲线，通常也是 $F^{d+1}, F^{d+2}, \cdots$ 连续的. 因此，F^n 连续性仅仅在曲线的维数 $d \geqslant n$ 时才有意义.

1.4.2 参数曲面间的连续性条件

对曲面间的连续性条件，有两种定义方式：参数连续及几何连续.

一个曲面 $\boldsymbol{p}(u,v)$ 在 (u_0,v_0) 处是 m 阶参数连续（记为 C^m）的充要条件是 $\mathrm{D}^a\boldsymbol{p}(u,v)$ 在 (u_0,v_0) 处皆连续，其中

$$\mathrm{D}^a\boldsymbol{p}(u,v) = \frac{\partial^{a_1}}{\partial u^{a_1}}\left(\frac{\partial^{a_2}}{\partial v^{a_2}}\boldsymbol{p}(u,v)\right).$$

$$a = \{a_1,a_2\}, \quad |a| = a_1 + a_2 \leqslant m, 0 \leqslant a_1, a_2 \leqslant |a|.$$

如果一个曲面片 $\boldsymbol{p}(u,v)$ 在区域 Ω 中的每一点皆是 C^m 的，则称曲面片 $\boldsymbol{p}(u,v)((u,v)\in\Omega)$ 关于参数 (u,v) 是 C^m 光滑的.

一个曲面片 $\boldsymbol{p}(u,v)$ 在 (u_0,v_0) 处 G^m 连续的充要条件是，$\boldsymbol{p}(u_0,v_0)$ 在其某个邻域内存在 C^m 参数表示. 下面给出严格的定义.

设 Δ_1 和 Δ_2 是平面上的两个区域，$\boldsymbol{E}_1(s)$、$\boldsymbol{E}_2(s)$ 分别是 Δ_1 和 Δ_2 的边界. 定义 $\boldsymbol{E}_1(s)$ 到 $\boldsymbol{E}_2(s)$ 的参数变换如下：

Φ 是定义在 $\boldsymbol{E}_1$ 的某一个邻域上的一个 C^m 同胚：$\Phi: U_{E_1(s)} \to V_{E_2(s)}$，$\Phi(\boldsymbol{E}_1(s)) = \boldsymbol{E}_2(s)$，并且 Φ 将 Δ_1 的内点映成 Δ_2 的外点，其中，$U_{E_1(s)}$ 是 $\boldsymbol{E}_1(s)$ 邻域，$V_{E_2(s)}$ 是 $\boldsymbol{E}_2(s)$ 的邻域. 设 Δ_1 和 Δ_2 是两个 C^m 曲面片，$\boldsymbol{p}_1:\Delta_1\to\mathbf{R}^3$ 和 $\boldsymbol{p}_2:\Delta_2\to\mathbf{R}^3$ 在 $\boldsymbol{E}_1$、$\boldsymbol{E}_2$ 处 G^m 光滑拼接的充要条件是，存在一个从 $\boldsymbol{E}_1$ 到 $\boldsymbol{E}_2$ 的 C^m 连接同胚 Φ，使得 $\boldsymbol{p}_1$、$\boldsymbol{p}_2\cdot\Phi$ 在 $\boldsymbol{E}_1(s)$ 上有直到 m 阶相同的偏导数，即

$$D^{\alpha}p_1 |_{E_1(s)} = D^{\alpha}(p_2 \cdot \Phi) |_{E_1(s)}, s \in [0,1], |\alpha| = 0,1,\cdots,m. \quad (1\text{-}17)$$

显然通过参数变换 Φ 得到了组合曲面在 $E_1(s)$ 的某一个邻域上的 C^m 参数表示：

$$p(x) = \begin{cases} p_1(x), & x \in \Delta_1, \\ p_2 \cdot \Phi(x), & x \bar{\in} \Delta_1, \Phi(x) \in \Delta_2. \end{cases}$$

从以上的定义不难看出，曲线的几何连续是它的特殊情形，如果 Δ_1、Δ_2 是 $\mathbf{R}^1$ 中的区间，则相应得到曲线的几何连续条件. 显然曲面的几何连续同参数选取无关，是一个仿射不变量.

1.4.3 有理曲线的连续性条件

对于有理参数曲线，如用齐次坐标表示的非有理曲线满足有关连续性约束，那么它的投影即有理曲线也同样具有相应的连续性，但反过来不成立. 如果有理曲线具有这些连续性，则相应它的齐次曲线并非一定要满足这些约束，即这些约束是充分而非必要的.

设有理曲线 $p(u)$ 是齐次曲线 $P(u) = (w(u)P(u), w(u))$ 在 $w = 1$ 超平面上的投影，且有理曲线 $p(u)$ 由在 u_0 处相接的两曲线段组成. 利用投影曲线的连续性条件，可以得到齐次曲线段间的约束关系.

1. 有理参数连续性约束

齐次曲线 $P(u)$ 的投影即有理曲线 $p(u)$ 在参数值 u 处是 C^n 连续的，当存在 $\alpha_i, i = 0,1,\cdots,n$，使得下列约束条件成立：

$$p_+^{(i)}(u) = \sum_{j=0}^{m} C_i^j \alpha_{i-j} p_-^{(i)}(u), \quad i = 0,1,\cdots,n.$$

写成矩阵形式为

$$\begin{bmatrix} p_+ \\ p'_+ \\ p''_+ \\ \vdots \\ p_+^{(n)} \end{bmatrix} = \begin{bmatrix} \alpha_0 & & & & \\ \alpha_1 & \alpha_0 & & & \\ \alpha_2 & 2\alpha_1 & \alpha_0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ C_n^0\alpha_n & C_n^1\alpha_{n-1} & C_n^2\alpha_{n-2} & \cdots & C_n^n\alpha_0 \end{bmatrix} \begin{bmatrix} p_- \\ p'_- \\ p''_- \\ \vdots \\ p_-^{(n)} \end{bmatrix}.$$

2. 有理几何连续性约束

几何连续性的贝塔约束(1-15)可写成形式

$$p_+^{(i)}(u) = \sum_{j=0}^{i} A_{i,j} p_-^{(j)}(u), \quad i = 0,1,\cdots,n.$$

利用它可导出当有理曲线 $p(u)$ 是 G^n 连续时，相应的齐次曲线 $P(u)$ 的一组线性约束条件是：齐次曲线 $P(u)$ 的投影曲线 $p(u)$ 是 G^n 连续的、即存在两组实数 α_i 与 β_i，使下式成立：

$$p_+^{(i)}(u) = \sum_{j=0}^{i} C_i^j \alpha_{i,j} \sum_{k=0}^{j} A_{j,k} p_-^{(k)}(u), \qquad i = 0,1,\cdots,n,$$

式中 C_i^j 是组合数，$\alpha_0 \neq 0$；

$$\begin{bmatrix} p_+ \\ p'_+ \\ p''_+ \\ p'''_+ \end{bmatrix} = \begin{bmatrix} \alpha_0 & & & \\ \alpha_1 & \alpha_0\beta_1 & & \\ \alpha_2 & \alpha_0\beta_2 + 2\alpha_1\beta_1 & \alpha_0\beta_1^3 & \\ \alpha_3 & \alpha_0\beta_3 + 3\alpha_1\beta_2 + 3\alpha_2\beta_1 & 3(\alpha_0\beta_1\beta_2 + \alpha_1\beta_1^2 + \alpha_0\beta_1^3) & \end{bmatrix} \begin{bmatrix} p_- \\ p'_- \\ p''_- \\ p'''_- \end{bmatrix}. \tag{1-18}$$

若 $\alpha_0 = 1$，其余所有 $\alpha_i = 0$，则有理几何连续性约束简化为贝塔约束. 另外、若 $\beta_1 = 1$，其余所有 $\beta_i = 0$，则有理几何连续性约束简化为有理参数连续性约束. (1-18) 式中右端的四阶关联矩阵就是(1-15) 和(1-17) 两式中关联矩阵左上角四阶子块的乘积.

1.5 有理参数多项式曲线与齐次坐标

采用有理参数多项式构造曲线、曲面，有两条优点：其一是有理参数多项式具有几何和透视投影不变性；其二是可以精确表示圆锥曲线、二次曲面，在几何造型系统中，便于统一曲线、曲面的数据结构和几何算法.

利用齐次坐标系可以把 n 维空间中的有理曲线表示成 $n+1$ 维空间中的多项式曲线.

设 $P = (x, y, z) \in \mathbf{R}^3$，则相应的 $(wx, wy, wz, w) = (X, Y, Z, w)$ 为四维齐次空间中点 P^w 的齐次坐标($w \neq 0$)，P 点可以通过中心在原点的透视变换，把 P^w 投影到 $w = 1$ 的超平面上得到. 记透视变换为 H，则

$$P = H(P^w) = H((X, Y, Z, w)) = \left(\frac{X}{w}, \frac{Y}{w}, \frac{Z}{w}\right).$$

若 $w = 0$，则 $(X, Y, Z, 0)$ 表示 $\mathbf{R}^3$ 中一个无穷远点，它代表了以 (x, y, z) 为方向的所有平行直线的公共点.

对于任意的 $x, y, z, w_1, w_2, w_1 \neq w_2$，有

$$\begin{aligned} H(P^{w_1}) &= H((w_1x, w_1y, w_1z, w_1)) = (x, y, z) \\ &= H((w_2x, w_2y, w_2z, w_2)) = H(P^{w_2}). \end{aligned}$$

例 1 三次参数多项式相应的有理多项式如下：

$$(wx, wy, wz, w) = (t^3, t^2, t, 1)\begin{bmatrix} a_x & a_y & a_z & a_w \\ b_x & b_y & b_z & b_w \\ c_x & c_y & c_z & c_w \\ d_x & d_y & d_z & d_w \end{bmatrix},$$

展开有

$$x(t) = \frac{a_xt^3 + b_xt^2 + c_xt + d_x}{a_wt^3 + b_wt^2 + c_wt + d_w},$$

$$y(t) = \frac{a_yt^3 + b_yt^2 + c_yt + d_y}{a_wt^3 + b_wt^2 + c_wt + d_w},$$

$$z(t)=\frac{a_z t^3+b_z t^2+c_z t+d_z}{a_w t^3+b_w t^2+c_w t+d_w}.$$

$x(t),y(t),z(t)$ 即为三次参数曲线的有理多项式.

对于几何式

$$\boldsymbol{p}=\boldsymbol{TMB},$$

其齐次坐标形式为

$$\boldsymbol{p}_w=\boldsymbol{TMB}_w,$$

其中

$$\boldsymbol{B}_w=(w_0\boldsymbol{p}_0,w_1\boldsymbol{p}_1,(w_0\boldsymbol{p}_0)',(w_1\boldsymbol{p}_1)').$$

由上面代数式的讨论可知，w 是参数 t 的函数，故

$$\boldsymbol{B}_w=(w_0\boldsymbol{p}_0,w_1\boldsymbol{p}_1,w'_0\boldsymbol{p}_0+w_0\boldsymbol{p}'_0,w'_1\boldsymbol{p}_1+w_1\boldsymbol{p}'_1),$$

这就是埃尔米特参数曲线几何式的有理形式.

类似地，也可构造其它参数曲线、曲面的有理形式.

1.6　参数曲线、曲面的重新参数化

1.6.1　参数曲线的重新参数化

参数曲线的重新参数化，就是重新确定曲线上点与参数域内的点之间的对应关系.对一条曲线的重新参数化，是为改变生成这条曲线的参数间隔，并不改变曲线的形状和位置.

设给定一正则曲线 $\boldsymbol{p}=\boldsymbol{p}(u),u\in[u_0,u_1]$，即对任意 $u\in[u_0,u_1]$，恒有 $\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}u}\neq 0$.若令 $u=u(t)$，满足 $\frac{\mathrm{d}u}{\mathrm{d}t}\neq 0$，且 $[t_0,t_1]\Rightarrow[u_0,u_1]$，则 $\boldsymbol{p}=\boldsymbol{p}(u(t))$，$t\in[t_0,t_1]$ 称为对曲线 $\boldsymbol{p}(u)$ 的重新参数化，$u=u(t)$ 为对曲线进行的参数变换.

由 $\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}=\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}u}\frac{\mathrm{d}u}{\mathrm{d}t}$ 可知，若 $\frac{\mathrm{d}u}{\mathrm{d}t}>0$，曲线取向不变；若 $\frac{\mathrm{d}u}{\mathrm{d}t}<0$，则取向相反.参数变换前后，曲线上一点的切向可能同向或者反向，由 $\frac{\mathrm{d}u}{\mathrm{d}t}$ 的正负决定. $\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}$ 与 $\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}u}$ 的模长相差 $\frac{\mathrm{d}u}{\mathrm{d}t}$ 倍.

常用的参数变换是线性变换.给定曲线段 $\boldsymbol{p}=\boldsymbol{p}(u),u\in[0,1]$，改变其参数域为 $t\in[t_1,t_2]$，则可作参数变换 $u(t)=\frac{t-t_1}{t_2-t_1},t_1\leqslant t\leqslant t_2$.反过来，若将参数域 $u\in[u_1,u_2]$ 变到 $t\in[0,1]$，则作变换 $u(t)=(1-t)u_1+tu_2,0\leqslant t\leqslant 1$.

曲线的重新参数化可用来重新界定曲线，如把曲线化成标准型，也常用于对曲线的分割(split)或裁剪(trimming).

若 $u(t)$ 是非线性函数，则

$$\frac{\mathrm{d}^2\boldsymbol{p}}{\mathrm{d}t^2}=\frac{\mathrm{d}^2\boldsymbol{p}}{\mathrm{d}u^2}\left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)^2+\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}u}\frac{\mathrm{d}^2u}{\mathrm{d}t^2}.$$

可见对新参数的二阶导矢$\frac{\mathrm{d}^2 p}{\mathrm{d}t^2}$与对原参数的二阶导矢$\frac{\mathrm{d}^2 p}{\mathrm{d}u^2}$相比，不仅方向发生改变，而且模长也发生了改变.曲线上点与参数域内点的对应关系也发生了改变.

用不同方程描述同一条曲线，其间差别在于曲线上的点与参数域内的点的对应关系不同.非线性变换会使曲线的次数升高.

曲线取自身的弧长为参数，称为**弧长参数化**.取弧长为参数的曲线上任一点，其切矢为单位矢量.由此可简化计算和公式推导.弧长参数化及以弧长的线性函数为参数的曲线参数化都是均匀的曲线参数化，即参数域内均匀分布的点对应曲线上沿曲线弧长均匀分布的点.值得注意的是，单一或分段的多次式曲线及有理多项式曲线，不可能取自身弧长为参数.

1.6.2 参数曲面的重新参数化

参数曲面的重新参数化，是确定曲面上的点与参数平面上参数域内的点之间的一种对应关系.

给定一正则曲面 $p = p(u,v),(u,v) \in R$，令

$$\begin{cases} u = u(\bar{u},\bar{v}), \\ v = v(\bar{u},\bar{v}), \end{cases} \qquad (\bar{u},\bar{v}) \in \bar{R}.$$

满足雅可比(Jaccobi)行列式不为零：

$$\frac{\partial(u,v)}{\partial(\bar{u},\bar{v})} = \begin{vmatrix} \frac{\partial u}{\partial \bar{u}} & \frac{\partial u}{\partial \bar{v}} \\ \frac{\partial v}{\partial \bar{u}} & \frac{\partial v}{\partial \bar{v}} \end{vmatrix} \neq 0,$$

则得到一个以 $\bar{u}$、$\bar{v}$ 为参数的曲面$p = p(u(\bar{u},\bar{v})),v(\bar{u},\bar{v}))(\bar{u},\bar{v} \in \bar{R})$.这一过程称为曲面的重新参数化.

雅可比行列式不为零，保证对曲面的变换是从正则到正则的.曲面的法矢为

$$\bar{p}_u \times \bar{p}_v = \frac{\partial(u,v)}{\partial(\bar{u},\bar{v})} p_u \times p_v.$$

用不同的方程描述同一张曲面，其差别在于曲面上的点与参数域内的点的对应关系不同.非线性的参数变换也将使曲面的次数升高.

利用参数变换，可以对曲面作分割，或者界定其中某一子曲面片.

1.7 曲线、曲面的光顺性

光顺是CAD/CAM和几何造型系统中十分重要的功能.关于光顺问题，主要集中在三个方面：①“光顺”概念的界定，即如何给出造型曲线、曲面是否“光顺”的准则.这一点需要具体的物理和几何背景，或者满足一定的“美学”准则，并且便于数学上的描述和计算机的实现；② 构造快速实用的光顺算法；③ 交互光顺造型设计.

1.7.1 光顺准则的定义

已有的光顺准则一般可分为两种类型：

(1) 整体能量型准则　　此类准则是由物理和几何背景出发而定义某种“能量”，来表征要光顺曲线、曲面的整体性质．在数学上是一个泛函问题，并认为使这一泛函达到极小的曲线、曲面是光顺的．

(2) 坏点判别准则　　此类准则都是从局部到整体，即对曲线、曲面上的点，给出一个判断“好”与“不好”的度量，同时给出一个相关整体好坏的度量．光顺的目标是使“不好”的点尽可能少，同时使整体度量变小．

对于平面曲线的光顺性定义，已有较成熟的光顺准则．下列几条光顺性定义是常用的．

定义 4　凡满足下列三条准则的平面曲线叫做光顺曲线：

1° 曲线 C^2 连续．

2° 没有多余拐点．

3° 曲率变化较均匀．

定义 5　一条曲线是光顺的，若其曲率是连续的，变号次数较少，而且尽可能近似于一个段数尽可能少的分段单调函数．

定义 6　一条曲线是光顺的，若其在满足给定插值要求的同时使得曲率平方对弧长的积分达到极小，即满足 $\min\int k^2\mathrm{d}s$．

在几何外形设计和构造中，在许多情况下几何外形信息是由型值点列(阵)给出，尤其在离散造型中，考察的大多数是数据点(阵)或控制网格等，因此需要考虑光顺的型值点列．

定义 7　对于平面上给定的一组型值点列$\{P_i\}_{i=1}^n$，若至少能找到一条光顺的插值曲线，就称$\{P_i\}_{i=1}^n$是光顺的．

一种曲面光顺准则是下列能量泛函：

$$\min\int_{\Omega}(k_1^2+k_2^2)\mathrm{d}\sigma,$$

式中 k_1、k_2 为曲面的主曲率，$\mathrm{d}\sigma$ 为单位面积元．

一般地，光顺度量的界定和光顺准则的选择，取决于几何外形的造型需要和曲线曲面的几何性态．J. Koulier 等人提出一种光顺准则的构造方法．

(1) 选择光顺趋势(平坦、圆滑等等)．

(2) 由要光顺的曲线、曲面得出诱导曲线、曲面，诱导方法是来自于(1) 所选定的趋势．

(3) 使诱导曲线、曲面的弧长或面积达到极小，此时的曲线、曲面是光顺的．

这种想法的特点在于构造出了一族光顺准则，以满足不同的造型需求，比如：对曲面来说，常用以下几种度量：

(1) 平坦度量(flattening metric)

$$\boldsymbol{C}(u,v)=K(u,v)\boldsymbol{n}(u,v).$$

(2) 圆滑度量(rounding metric)

$$\boldsymbol{C}(u,v)=\boldsymbol{r}(u,v)+[H(u,v)/K(u,v)]\boldsymbol{n}(u,v).$$

(3) Rolling 度量

$$C(u,v) = [K(u,v) + H^2(u,v)]n(u,v).$$

光顺的目标是使 $C(u,v)$ 的面积达到极小.

一种曲面的光顺方法是光顺其相应的网格线,与曲线情形类似的曲面光顺准则是:

(1) 关键曲线光顺,如飞机或轮船的骨架线.

(2) 高斯曲率变化均匀.

(3) 网格线光顺.

需要指出的是,曲线、曲面边界条件的确定,对曲线、曲面的光顺性也有较大的影响.

1.7.2 几种常用的光顺方法

光顺的方法也可分为两大类:整体化方法,选点修改方法.

1. 整体化方法

此类方法主要针对"能量"准则,采用各种优化的方法来解光顺问题,如董光昌等提出的参数优化法、回弹法、能量法、最小二乘法等.

整体化方法的共同点是:依赖于优化算法的好坏,计算量比较大,收敛速度不够快.整体光顺并不一定能保证局部也是最佳的,还可能出现"光顺型值"点振荡的情况.

(1) **参数优化法** 已知插值点列 $\{P_i\}_{i=1}^m$,位置矢量 p_i, $i = 1,2,\cdots,m$,和 k 次 B 样条的节点序列 $\{u_i\}$, $1 \leqslant i \leqslant m+k+1$,选择 t_i, $i = 1,2,\cdots,m$,满足

$$t_1 = u_{k+1},\ t_m = u_{m+1},\ u_i < t_i < u_{i+k+1},\ 2 \leqslant i \leqslant m-1, \tag{1-19}$$

从而存在唯一的控制点列 d_i, $i = 1,2,\cdots,m$,使得

$$C(t_i) = \sum_{j=1}^{m} d_j N_{j,k}(t_i), \qquad i = 1,2,\cdots,m \tag{1-20}$$

这里 $N_{j,k}(t)$ 为以 $\{u_i\}$ 为节点的 B 样条基.

参数 $t_i (i = 1,2,\cdots,m)$ 的选取会直接影响最终得到的 B 样条曲线 $C(t)$ 的形状,为了得到满意的插值曲线,还必须对参数值 $t_i (i = 1,2,\cdots,m)$ 进行优化,为此引入衡量曲线光顺性的能量函数:

$$E_r(t_1,t_2,\cdots,t_m) = \int_{t_1}^{t_m} \| C^{(r)}(t) \|^2 dt \tag{1-21}$$

作为参数优化的目标函数.求得满足条件(1-21)的 $t_i,\cdots,t_m$,使得 $E_r(t_i,\cdots,t_m)$ 取极小值,从而由(1-20)唯一求得 B 样条曲线 $C(t)$ 便是光顺插值曲线.利用最速下降法,可以求得 $E_r(t_i,\cdots,t_m)$ 的极小值,其中每一步的下降方向为

$$\boldsymbol{v} = -\left(0, \frac{\partial E_r}{\partial t_2}, \cdots, \frac{\partial E_r}{\partial t_{m-1}}, 0\right).$$

如果在某型值点处加上切矢 p'_i 和二阶导矢 p''_i, $i \in S \subset \{1,2,\cdots,m\}$,作为插值约束条件,那么对于取定的参数节点 u_j, $1 \leqslant j \leqslant n+k+1$, $n = m + m_1 + m_2$,以及对应每个型值点的参数值 t_i, $1 \leqslant i \leqslant m$,插值 B 样条的控制点 d_j, $j = 1,2,\cdots,n$,应由下面的方程组唯一确定

$$\sum_{j=1}^{n} d_j N_{j,k}(t_i) = p_i, \quad 1 \leqslant i \leqslant m,$$

$$\sum_{j=1}^{n} d_j N_{j,k}^{(1)}(t_i) = a_i p'_i, \quad i \in S_1,$$

$$\sum_{j=1}^{n} d_j N_{j,k}^{(2)}(t_i) = a_i^2 p'_i, \quad i \in S_2,$$

其中 a_i 为正数，$i \in S_i \cap S_2$. 类似地，可以取前面定义的能量函数(1-21)作为目标函数，把$\{t_i\}$、$\{a_i\}$作为变量，通过迭代法求得目标函数的极小值，从而达到光顺拟合的目的.

(2) **回弹法**　回弹法是手工放样中的"两借借，自然放"的一种数学模拟，通过新老两组型值点交替固定和回弹，使样条的能量渐次减少，以达到光顺的目的.

当平面上给定一组型值点 $P_i(x_i, y_i)(i = 0,1,\cdots,n)$ 以及适当的边界条件后，回弹法的计算步骤如下所述：

步1　对于给定的分割

$$\Delta: a = x_0 < x_1 < \cdots < x_n = b$$

及型值点$\{P_i(x_i, y_i)\}$，作插值三次样条曲线 $S(x)$.

步2　取相邻两个节点的中点

$$\xi_i = \frac{1}{2}(x_i + x_{i+1}) \quad (i = 0,1,\cdots,n-1),$$

记 $\xi_{-1} = a, \xi_n = b$，而且作插值 $S(\xi_i)$.

步3　对于另一分割

$$\Delta^*: a = \xi_{-1} < \xi_0 < \cdots < \xi_{n-1} < \xi_n = b$$

及型值点$\{Q_i(\xi_i, S(\xi_i))\}(i = 0,1,\cdots,n)$，同样作插值三次样条曲线 $S^*(x)$.

步4　称 $P_i^*(x_i, S^*(x_i))(i = 0,1,\cdots,n)$ 为经过一次回弹的新型值点.

步5　如此重复，直到某一次回弹前后的节点 x_i 处的函数值之差小于定值ϵ为止，即

$$|S^*(x_i) - S(x_i)| < \epsilon \qquad (i = 0,\cdots,n).$$

最后一次获得的型值点及其插值三次样条即作为光顺型值点和光顺曲线. 在船体数学放样中，一般取 $\epsilon = 3\text{mm}$.

当构造插值三次样条曲线 $S(x)$ 和 $S^*(x)$ 时，它们的边界条件在回弹过程中不变. 第2步的插节点可以改成加权平均形式

$$\xi_i = a_i x_i + (1 - a_i) x_{i+1} \qquad (i = 0,\cdots,n-1),$$

其中权因子 $a_i \in (0,1)$，选择原则是使得$\dfrac{a_i}{1-a_i}$与样条$S(x)$在 x_i、x_{i+1} 两节点处的剪力跃度成比例.

直观地看，经过回弹，样条的能量逐次减少，曲线也就趋向光顺，回弹法可以看成一种迭代逼近的能量法. 但是，如果迭代次数过多的话，可能出现光顺型值点与原型值点偏离过大的问题，而且，对平直段小波动往往难以消除，这是整体光顺法

的通病.

(3) **最小二乘法** 在平面上给定了一组用坐标表示的型值点 $P_i(i=0,1,\cdots,n)$,关于型值点的一种参数化分割为

$$\Delta: a = t_0 < t_1 < \cdots < t_n = b,$$

在分割 Δ 上的三次样条曲线为

$$S(t) = \sum_{i=0}^{3} a_i t^i + \sum_{i=1}^{n-1} b_i F(t), \quad t \in [a,b], \tag{1-22}$$

式中,$F(x)$ 是定义在分割Δ上的样条基函数,取截断幂级数表示:$(x-x_i)_+^3$ 或者取 B 样条基函数等. $n+3$ 个系数 $\{a_i\}$ 和 $\{b_i\}$ 唯一地决定下述目标函数 I 的极小,从而得到一条逼近地通过原型值点 $\{P_i\}$ 的三次样条 $S(t)$. $S(t)$ 所代表的曲线称为**最小二乘光顺曲线**,点 $P_i^*(t_i, S(t_i))(i=0,\cdots,n)$ 称为光顺型值点.

一种目标函数是

$$I = \sum_{i=0}^{n} \alpha_i [S(t_i) - p_i]^2 + \sum_{i=1}^{n-1} \beta_i b_i^2, \tag{1-23}$$

式中 $2n$ 个正权因子 $\alpha_i(i=0,1,\cdots,n)$ 和 $\beta_i(i=0,1,\cdots,n-1)$ 是事先予以给定的.

1° 当取所有的 $\beta_i=0, \alpha_i \neq 0$ 时,所有光顺的型值点 $\{P_i^*\}$ 合同于原型值点 $\{P_i\}$.

2° 当取所有的 $\alpha_i=0, \beta_i \neq 0$ 时,(1-22) 式成为一个整体三次多项式

$$S(t) = \sum_{i=0}^{3} a_i t^i,$$

$S(t)$ 在各内节点 t_i 处的剪力跃度 b_i 全部为零. 按照样条的力学模型,剪力跃度 b_i 代表样条在型值点 P_i 处的回弹力,所以这时的样条 $S(t)$ 表示了一条最光顺的曲线.

当 $\{\alpha_i\}$ 取大数时,光顺后的型值点 $\{P_i^*\}$ 与原型值点 $\{P_i\}$ 的偏离就很小,剪力跃度 $\{b_i\}$ 也就很小,曲线也就光顺,但 $S(t)$ 与 P_i 的偏离或许会较多,以致曲线的逼近度较差. 因此,称 $\{\alpha_i\}$ 为**偏离权**,$\{\beta_i\}$ 为**光顺权**. 极小化目标函数 I 的问题归结为 $n+3$ 个未知数 $a_0,\cdots,a_3; b_1,\cdots,b_{n-1}$ 的线代数方程组求解,即

$$\begin{cases} \dfrac{\partial I}{\partial a_i} = 0 \quad (i=0,\cdots,3), \\ \dfrac{\partial I}{\partial b_j} = 0 \quad (j=1,\cdots,n-1). \end{cases} \tag{1-24}$$

可以证明,方程组(1-24) 的解唯一存在. 若三次样条函数取截断幂级数为基底,当 n 增大时,方程组(1-24) 的系数矩阵的病态程度将急剧增加,引起计算的不稳定. 若取三次 B 样条函数作为基底,便不会出现这个问题. 而且,这时方程组 (1-24) 的系数矩阵呈现七对角的带状结构,求解简单.

(4) **能量法** 由日本学者穗板卫提出,假设拟合曲线是局部小挠度的,它便于曲线上弧长和曲率计算的线性化处理. 它的基本思想同最小二乘法一样,是偏离和光顺两部分的加权综合. 不同之处在于:

1° 采用累加弦长三次参数样条作为拟合曲线.

2° 目标函数中的剪力跃度部分改成样条的能量积分.

在空间给定一组待光顺的型值点 $Q_i(i=0,\cdots,n)$. 过光顺后的型值点 $P_i(i=0,1,\cdots,n)$ 作一条插值的弹性线,而且在 P_i 与 Q_i 两点间挂一条弹性系数为 α_i 的小弹簧.这样,包括弹性线和小弹簧在内的整个系统的内能

$$U=\frac{1}{2}\sum_{i=0}^{n}\alpha_i(P_i-Q_i)^2+\frac{1}{2}(EI)^2\int k^2\mathrm{d}s, \tag{1-25}$$

式中常数 EI 为弹性线的刚度.

能量法容易推广到二维网格的光顺问题.

2. 选点修改法

这是当前比较流行的方法,即按照光顺准则判别局部最"坏"的点,加以修正、迭代、反复调整,以达到由局部修改而带动整体光顺的效果.此类方法的优点是修改能力强、收敛速度快,并且容易将方法局部化.选点修改法的基础是,承认极大多数型值点是好的或比较好的,选点修改法的光顺过程就是把少数"坏点"挑出来逐个予以修正.

选点修改法的优点是:① 坏点挑得准,好点不受损,修改的型值点少.② 严格满足三条光顺准则,圆满解决了平直段小波动问题.③ 修改能力强.

选点修改法的缺点是:当连续出现多个坏点时,往往不容易处理好.虽然也能光顺,收敛速度就慢了.

选点修改法之所以特别适用于几何外形的光顺问题,是因为在这类问题中有极大多数的好点.而在数据拟合问题中,因计算误差和系统误差的影响,往往难以明确区分好点和坏点,对于这类问题,似乎采用整体光顺拟合法为好.

G. Farin 等人针对三次 B 样条曲线提出自动光顺算法:

(1) 坏点判别准则　跃度 K' 最大的点,K 表示曲线的曲率;

(2) 坏点修改　采用 B 样条的节点移去算法,使 k' 从不连续到连续;

(3) 整体光顺度量　$\sum|K'_+-K'_-|$.

复旦大学数学系和江南造船厂提出的基样条法与山东大学和沪东造船厂提出的圆率法都是选点修改法.

(1) **基样条法**　利用曲线光顺性定义的三条准则,基样条法光顺步骤如下:

步 1　对于给定的原始型值点 $P_i(x_i,y_i)(i=0,1,\cdots,n)$ 和边界导数 y'_0、y'_n 构造插值三次样条函数 $S(x)$.

步 2　计算 $S(x)$ 在每个节点 x_i 处的二阶导数 $M_i=S''(x_i)$,并作符号序列 $\{\mathrm{sign}(M_i)\}$.在点列中,凡使符号序列连续变号的点称为坏点.光顺的目标是使得符号序列无连续变号.这一步称为初光顺,它具有消除多余拐点的功能.

步 3　作二阶导数差分 $\Delta M_i=M_i-M_{i-1}$ 的符号序列 $\{\mathrm{sign}(\Delta M_i)\}$.在点列中,凡使差分符号序列连续变号的点称为坏点,光顺的目标是使得差分符号序列无连续变号.这一步称为精光顺,它具有均匀化曲线曲率变动的功能.

一般地,设 $P_k(x_k,y_k)$ 是坏点,而且修改后的点为 $P_k^*(x_k,y_k+\rho_k)$.现仅改动第

k 个点 P_k 而保持其余各点不变，记所得的插值三次样条为 $S^*(x)$，那么，成立关系式：

$$S^*(x) = S(x) + \rho_k\varphi_k(x) \tag{1-26}$$

式中 $\varphi_k(x)$ 是样条基函数.

在基样条法中所希望的是，修改后的样条 $S^*(x)$ 的剪力跃度 $\{b_i^*\}$ 的平方和取极小.

光顺的过程就是对挑出的全部坏点逐点进行修改. 先进行初光顺，然后，等到型值点序列全部满足光顺条件(2)，再进行光顺，使之满足光顺条件(3).

基样条法可以推广到曲面情形.

(2) **圆率法** 圆率法也是一种选点修改法，它不需要插值曲线，而从离散型值点分布的几何位置出发直接判断型值点列的光顺性，进而挑出坏点给以光顺修改.

在平面上给定了型值点列 $P_i(i = 0,1,\cdots,n)$ 和两边界切向 m_0, m_n，过相邻三点 P_{i-1}、P_i、P_{i+1} 所作圆的相对曲率 K_i 称为在点 P_i 处的圆率. 当圆弧 $P_{i-1}P_iP_{i+1}$ 走向为逆时针时，K_i 取正号，顺时针时则取负号. 边界点 P_0 处的圆率 K_0 则以过两点 P_0、P_1 和 P_0 处的切向 m_0 所作的圆来确定，K_n 亦然，这样，便得到对应于型值点列 $\{P_i\}$ 的圆率序列 $\{K_i\}$.

圆率法可分成初光顺和精光顺两部分：

1° 初光顺 在符号序列 $\{\mathrm{sign}(K_i)\}$ 中，凡使连续变号的点为坏点. 初光顺的目标是要达到圆率符号序列无连续变号.

2° 精光顺 在圆率差分 $\Delta K_i = K_i - K_{i-1}$ 的符号序列 $\{\mathrm{sign}(K_i)\}$ 中，凡使连续变号的点为坏点. 精光顺的目标是要达到圆率差分符号序列无连续变号.

圆率法采用圆率的二次差变成最小作为光顺目标.

定义 8 P_i 点处圆率的二次差是指

$$D_i = \lambda_i K_{i-1} + \mu_i K_{i+1} - K_i,$$

式中

$$\lambda_i = \frac{l_{i+1}}{l_i + l_{i+1}}, \mu_i = \frac{l_i}{l_i + l_{i+1}}, l_i = \overline{P_{i-1}P_i}.$$

假定 P_i 是初光顺中的坏点，那么圆率 K_{i-1} 与 K_{i+1} 为同号，而与 K_i 异号. 因此 $|D_i|$ 较大. 这样，圆率的二次差的绝对值 $|D_i|$ 是能相对地反映出 P_i 点邻近的光顺程度的.

将坏点 P_i 修改成光顺点 P_i^* 的办法如下：假定在原型值点列中仅用 P_i^* 替代 P_i，而保持其余各点不变，记新型值点列的圆率序列为 $\{K_i^*\}$. 修改的原则是使得 $\{K_i^*\}$ 在 P_i^* 处圆率的二次差

$$D_i^* = 0$$

圆率法的实质在于：通过对圆率二次差 D 的减少而起到减少剪力跃度 b 的作用，从而它具有光顺曲线的功能.

平面曲线的光顺性，常用曲率半径线来检查.

利用曲面的测地线、曲率线、等高线等的绘制(如高光法(highlight))，或用曲面

的高斯曲率的颜色值绘制(称为上色法),分析曲面的形状和光顺性,以利于曲面光顺性的交互修改.

1.8 张量积曲面

设三维欧氏空间中的曲面表示为

$$S(u,v) = (x(u,v), y(u,v), z(u,v)),$$

参数$(u,v) \in D$,D是由u、v构成的平面参数域.参数曲面有多种表示形式,其中最简单的、应用最广泛的表示形式是张量积(tensor product)形式.

张量积形式是关于u、v的双变量函数,它是两个简单变量基函数的乘积,几何系数是双向排列的$n \times m$网络点.从而张量积曲面的表示形式为

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} f_i(u) g_i(v) b_{ij}, \tag{1-27}$$

式中,$b_{ij} = (x_{ij}, y_{ij}, z_{ij})$,$0 \leqslant u$、$v \leqslant 1$,参数域$(u,v)$是矩形域.$S(u,v)$也可以写成矩阵形式

$$S(u,v) = [f_i(u)]^{T}[b_{ij}][g_j(v)]$$

式中$[f_i(u)]^{T}$是$1 \times (n+1)$行向量,$[g_j(v)]$是$(m+1) \times 1$列向量,$[b_{ij}]$是$(n+1) \times (m+1)$三维点阵.

例2 幂基曲面表示如下:

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} a_{ij} u^i v^j = [u^i]^{T}[a_{ij}][v^j], \tag{1-28}$$

式中$0 \leqslant u$、$v \leqslant 1$. $f_i(u) = u^i$, $g_i(v) = v^j$.

例3 贝齐尔曲面由双向控制网格点和两个单变量的伯恩斯坦多次式的乘积组合而成:

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{ij} B_{i,n}(u) B_{j,m}(u), \quad 0 \leqslant u、v \leqslant 1, \tag{1-29}$$

固定参数$u = u_0$,

$$\begin{aligned} C(v)|_{u_0} = S(u_0, v) &= \sum_{i=0}^{n} \sum_{j=0}^{m} p_{ij} B_{i,n}(u_0) B_{j,m}(v) \\ &= \sum_{j=0}^{m} \left(\sum_{i=0}^{n} p_{ij} B_{i,n}(u_0) \right) B_{j,m}(v) \\ &= \sum_{j=0}^{m} Q_j(u_0) B_{j,m}(v) \end{aligned}$$

式中$Q_j(u_0) = \sum_{i=0}^{n} p_{ij} B_{i,n}(u_0)$, $j = 0,1,\cdots,m$. $C(v)|_{u_0}$是曲面上的贝齐尔曲线,是参数曲线,或称v线.类似有u线$C(u)|_{v_0}$,两参数线$C(u)|_{v_0}$和$C(v)|_{u_0}$交于曲面上一点$S(u_0, v_0)$.

2　常用的参数曲线、曲面

2.1　三次参数样条曲线、曲面

在参数样条曲线中，三次样条曲线最常用. 这是因为 ① 这是次数最低的 C^2 类样条，可满足大多数工程和数学物理的需要. 次数低则带来计算的简便和稳定；② 是木样条的线性近似.

2.1.1　三次参数曲线段

设曲线的表示为

$$\boldsymbol{p}(t) = \boldsymbol{p}_0 + \boldsymbol{p}_1 t + \frac{1}{2}\boldsymbol{p}_2 t^2 + \frac{1}{6}\boldsymbol{p}_3 t^3, \qquad t \in [0,1], \tag{2-1}$$

则称它为**三次参数曲线段**，可以证明，参数 t 不可能是曲线的弧长参数.

三次参数曲线段上可能出现多余的拐点和奇点. 下面考察平面三次参数曲线的仿射不变量，对于曲线(2-1)，设 $\boldsymbol{p}_i = (a_i, b_i)$，$i = 0,1,2,3$，它的分量形式是

$$\begin{cases} x = a_0 + a_1 t + \dfrac{1}{2}a_2 t^2 + \dfrac{1}{6}a_3 t^3, \\ y = b_0 + b_1 t + \dfrac{1}{2}a_2 t^2 + \dfrac{1}{6}b_3 t^3. \end{cases}$$

记

$$p = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix}, q = \begin{vmatrix} a_3 & a_1 \\ b_3 & b_1 \end{vmatrix}, r = \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix}$$

方程

$$\frac{1}{2}pt^2 - qt + r = 0$$

称为曲线(2-1)的**拐点方程**.

定理1　设平面三次参数曲线的方程是(2-1)式，$t \in \mathbf{R}$. 当 $p = 0$ 时，它没有二重点和尖点，只有一个拐点；当 $p \neq 0$ 时，按照 $q^2 - 2pr$ 的符号分成三种情形：

1° 若 $q^2 - 2pr > 0$，它仅有两个实拐点，$t = \dfrac{q}{p} \pm \sqrt{\dfrac{q^2 - 2pr}{p}}$；

2° 若 $q^2 - 2pr = 0$，仅有一个尖点，$t = \dfrac{q}{p}$；

3° 若 $q^2 - 2pr < 0$，仅有一个二重点，而没有实拐点.

利用曲线段端点的条件，可以使(2-1)式的几何意义更明显.

设 $\boldsymbol{p}(0)$、$\boldsymbol{p}(1)$、$\boldsymbol{p}'(0)$、$\boldsymbol{p}'(1)$ 分别表示(2-1)式两端点的位置值和切矢，则(2-1)式可以写成如下形式：

$$\boldsymbol{p}(t) = \boldsymbol{p}(0)F_0(t) + \boldsymbol{p}(1)F_1(t) + \boldsymbol{p}'(0)G_0(t) + \boldsymbol{p}'(1)G_1(t), \tag{2-2}$$

其中 $F_i(t)$、$G_i(t)(i=0,1)$ 为三次埃尔米特调配函数.(2-2) 式也称为曲线(2-1)的埃尔米特插值形式.

2.1.2 三次参数样条曲线

定义 1 已知空间中有序的 $n+1$ 个点 $P_0, P_1, \cdots, P_n$,用参数三次曲线段连接相邻两点,如果由这几段曲线段组成的曲线在各正则点的切线和曲率向量都是连续的,则称这条曲线为一般的**三次参数样条曲线**.

当参数分割 $\Delta u: u_0 < u_1 < \cdots < u_n$ 确定后,三次参数样条曲线就由 $n+1$ 个数据点 $P_i, i=0,1,\cdots,n$ 及两个边界条件完全定义,因此总体上共有 $n+3$ 个自由度.三次参数样条曲线的各个坐标分量都是定义在同一参数分割 Δu 上的三次样条函数,在分割 Δu 上的三次样条函数的全体构成 $n+3$ 维线性空间,其中任一组 $n+3$ 个线性无关的三次样条函数都可作为一组基.因此根据函数的不同形式,可以构造不同表示形式的三次参数样条曲线,例如:埃尔米特基函数(2-2) 形式、幂基形式、基样条形式、伯恩斯坦基形式、B 样条基形式等,其中最为常用的是后两种,但它们不是插值而是逼近形式,主要应用于几何外形设计.

三次样条曲线的形状还取决于对插值数据点的参数化方法,即参数分割方式的选择.参数分割通常有以下几种类型:均匀参数、累加弦长参数、向心参数、修正弦长参数等几种形式,其中累加弦长参数形式最为常见,常用于对大挠度曲线的拟合.

2.1.3 累加弦长三次参数样条曲线

设在相邻两个型值点 P_{i-1} 和 P_i 之间的第 i 段曲线方程为

$$p(t) = r_0 + r_1 t + r_2 t^2 + r_3 t^3, \qquad t_{i-1} \leqslant t \leqslant t_i \tag{2-3}$$

其中弦长 $l_i = \overline{P_{i-1}P_i}$,参数轴上的节点坐标

$$t_i = \sum_{j=1}^{i} l_j \quad (i=1,2,\cdots,n).$$

相邻两曲线段在型值点 P_i 处保持一阶和二阶导矢连续,并分别记为 m_i 和 M_i:

$$\begin{cases} p'(t_i-0) = p'(t_i+0) = m_i, \\ p''(t_i-0) = p''(t_i+0) = M_i. \end{cases}$$

利用各型值点 P_i 处的一阶导矢 m_i 和二阶导矢 M_i 之间的关系式

$$\begin{cases} l_i M_i = -2(3e_i - m_{i-1} - 2m_i), \\ l_{i+1} M_i = 2(3e_{i+1} - 2m_i - m_{i+1}), \end{cases}$$

得到 m 连续方程

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = 3(\lambda_i e_i + \mu_i e_{i+1}) \quad (i=1,2,\cdots,n-1), \tag{2-4}$$

式中 $\lambda_i = \dfrac{l_{i+1}}{l_i + l_{i+1}}, \mu_i = \dfrac{l_i}{l_i + l_{i+1}}, e_i = \dfrac{1}{l_i}(P_i - P_{i-1})$ 表示相邻两型值点 P_{i-1} 和 P_i 所连弦方向上的单位向量.

相应地有 M 连续方程

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = 6\frac{e_{i+1} - e_i}{l_i + l_{i+1}} \qquad (i = 1,2,\cdots,n-1). \tag{2-5}$$

连续性方程(2-4)或(2-5)添加两个适当的边界条件后，可完全确定 m_i 或 $M_i(i = 1,2,\cdots,n-1)$. 解线性方程组可利用追赶法，系数矩阵是严格对角占优的.

利用方程(2-4)或(2-5)，可以构造三次参数样条曲线，例如(2-2)式可写成

$$p(t) = p_{j-1}F_0\left(\frac{t - t_{j-1}}{l_j}\right) + p_jF_1\left(\frac{t - t_{j-1}}{t_j}\right) +$$
$$l_{j-1}m_{j-1}G_0\left(\frac{t - t_{j-1}}{l_j}\right) + l_jm_jG_1\left(\frac{t - t_{j-1}}{l_j}\right).$$

补充边界条件可选择：切矢条件或自由端点(零曲率)条件或抛物线条件等.

2.1.4　各种连接条件下的三次参数样条曲线

定理2　相连接的两条三次参数曲线段 $p_1(t)$ 和 $p_2(t)$，$t \in [0,1]$，在连接点处的切向和曲率都连续的充要条件是，存在常数 $\alpha(>0)$ 和 β_1 使得

$$\begin{cases} p_1(1) = p_2(0), \\ p'_1(1) = \alpha p'_2(0), \\ p''_1(1) = \alpha^2 p''_2(0) + \beta p'_2(0). \end{cases} \tag{2-6}$$

(2-6)式表明，具有 C^2 连续条件的最一般参数样条曲线并不唯一确定，只要适当调节两个自由度 α、β，便可构造出适合各种需要的样条曲线.

(1) 取所有的 $\alpha = 1, \beta = 0$，所得的是弗格森形式的三次参数样条曲线.

(2) 取 $\alpha_i = \dfrac{l_i}{l_{i+1}}, \beta_i = 0(i = 1,2,\cdots,n-1)$，便是累加弦长三次参数样条曲线.

(3) 取 $\alpha_i = \dfrac{l_i}{l_{i+1}}$，$\beta_i$ 作为张力参数$(i = 1,2,\cdots,n-1)$，得适合大挠度情形下 C^2 连续的张力样条曲线.

另外，利用 α、β 可构造保形插值样条曲线和规范样条曲线等.

2.1.5　参数样条曲面

参数曲面常用的是双三次曲面，常见的参数曲面有埃尔米特参数曲面，孔斯曲面，常见的参数样条曲面是张量积的贝齐尔曲面，B样条曲面和NURBS曲面等.

双三次参数曲面片的代数形式是

$$p(u,w) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}u^iw^j, \qquad u,w \in [0,1].$$

其矩阵表示为

$$p = UAW^{\mathrm{T}} \tag{2-7}$$

此处

$$U = [u^3\ u^2\ u\ 1],$$
$$W = [w^3\ w^2\ w\ 1],$$

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

双三次参数曲面片的几何表示是基于其代数表示和边界条件. 如图 2-1 所示,

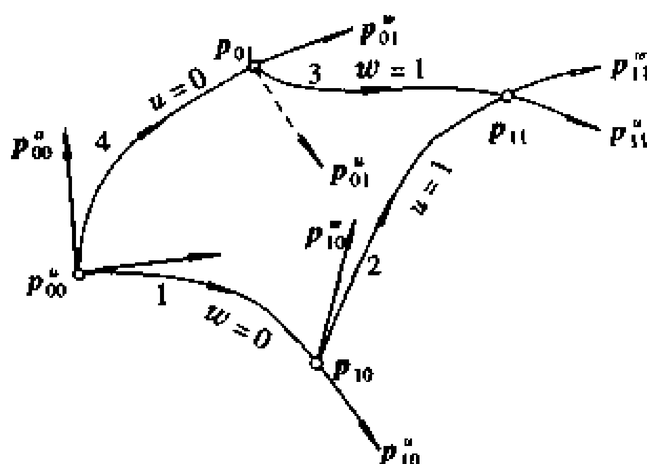


图 2-1

曲面片四个角点处的值及偏导数、混合导数的值分别为 $p_{ij}, p_{ij}^{u}, p_{ij}^{w}, p_{ij}^{uw}, i,j = 0,1$.
双三次参数曲面片具有如下形式的矩阵表示

$$p(u,w) = [F_0(u)\ F_1(u)\ G_0(u)\ G_1(u)] \begin{bmatrix} p_{00} & p_{01} & p_{00}^{w} & p_{01}^{w} \\ p_{10} & p_{11} & p_{00}^{w} & p_{11}^{w} \\ p_{00}^{u} & p_{01}^{u} & p_{00}^{uw} & p_{01}^{uw} \\ p_{10}^{u} & p_{11}^{u} & p_{10}^{uw} & p_{11}^{uw} \end{bmatrix} \begin{bmatrix} F_0(w) \\ F_1(w) \\ G_0(w) \\ G_1(w) \end{bmatrix}$$

$$= F(u)BF^{T}(w),$$

其中 $F(u) = [F_0(u)\ F_1(v)\ G_0(u)\ G_1(u)]$ 是埃尔米特调配函数.

令 $F(u) = UM, F(w) = WM$, 则三次参数曲面片的几何表示的矩阵式是

$$p = UMBM^{T}W^{T}, \quad u,w \in [0,1], \tag{2-8}$$

此处的 $U = [u^3\ u^2\ u\ 1], W = [w^3\ w^2\ w\ 1]^{T}$,

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

由(2-8)式定义的双三次参数曲面又称之为埃尔米特曲面或弗格森曲面. 基于(2-7)和(2-8)两式可以得到双三次参数曲面代数形式和几何形式之间的关系:

$$A = MBM^{T},$$

通常情况下构造参数曲面的主要任务就是构造它的几何系数矩阵 B.

2.2 贝齐尔曲线

2.2.1 贝齐尔曲线的定义及性质

n 次贝齐尔曲线定义为

$$p(u) = \sum_{i=0}^{n} b_i B_{i,n}(u), \quad 0 \leqslant u \leqslant 1. \tag{2-9}$$

式中$\{B_{i,n}(u)\}$是 n 次伯恩斯坦基函数 . $\{b_i\}$ 称为控制顶点,依次连接 b_i 构成的多边形,称为控制多边形或特征多边形 . 贝齐尔曲线可看作是对控制多边形的逼近 .

由伯恩斯坦基函数的性质,可得贝齐尔曲线的下列性质:

(1) **端点性质** $p(0) = b_0, p(1) = b_n$,并且在首末端点处的 k 阶导矢分别与控制多边形的首末 k 条边有关,与其它边无关 .

(2) **几何不变性与仿射不变性** 贝齐尔曲线的这种表示形式不依赖于坐标系的选取 . 在仿射变换下,曲线(2-9) 的像曲线是$\{b_i\}$ 的像所决定的贝齐尔曲线 .

(3) **对称性** 设 $b_i^* = b_{n-i}(i = 0,1,\cdots,n)$,由此生成的新的贝齐尔曲线

$$p^*(u) = \sum_{i=0}^{n} b_i^* B_{i,n}(u) = \sum_{i=0}^{n} b_i B_{i,n}(1-u) = p(1-u).$$

(4) **凸包性质** 对于 $u \in [0,1], p(u) = \sum_{i=0}^{n} b_i B_{i,n}(u) \in T$,这里 T 为$\{b_i\}$ 的凸包,

$$T = \left\{ \sum_{i=0}^{n} \lambda_i b_i \,\middle|\, \sum_{i=0}^{n} \lambda_i = 1, \lambda_i \geqslant 0, i = 0,1,\cdots,n \right\}.$$

(5) **变差缩减性**(variation diminishing) 除包含整个控制多边形的平面外,任一平面与贝齐尔曲线的交点数不超过它与控制多边形的交点数,由此可得平面贝齐尔曲线凸性定理:在平面上,若贝齐尔控制多边形是凸的,则其贝齐尔曲线也是凸的 .

(6) 移动 n 次贝齐尔曲线(2-9) 的第 j 个控制顶点 b_j,将对曲线上参数为 $u = \frac{j}{n}$ 的那个点$p\left(\frac{j}{n}\right)$发生最大影响 .

2.2.2 几何作图法 —— 德卡斯特里奥算法

对于给定的控制点 $b_i(i = 0,1,\cdots,n)$,给定 $t \in [0,1]$,记 $b_i^{(0)}(t) = b_i, i = 0, 1,\cdots,n$. 在至 $b_i^{(0)}(t)$ 线段的上取一点 $b_i^{(1)}(t)$,使得

$$b_i^{(1)}(t) = (1-t) b_i^{(0)}(t) + t b_{i+1}^{(0)}(t),$$

从而得到 n 个点$b_i^{(1)}(i = 0,1,\cdots,n-1)$;再在以 $b_i^{(1)}(t)$ 为顶点的多边形上进行如上操作,则得到 $n-1$个点 $b_i^{(2)}(t)(i = 0,1,\cdots,n-2)$;这样下去,第 r 次得$n-r+1$个点:

$$b_i^{(r)}(t) = (1-t)b_i^{(r-1)}(t) + tb_{i+1}^{(r-1)}(t), \quad i = 0,1,\cdots,n-r. \tag{2-10}$$

当 $r = n$ 时,只剩下一个点 $b_0^{(n)}(t)$,则 $b_0^{(n)}(t)$ 为 n 次贝齐尔曲线在 t 处的点. 图 2-2 即为 $n = 3, t = 1/4$ 时的几何作图过程.

对任意给定的 $b_i(i = 0,1,\cdots,n)$ 及固定的 $t \in [0,1]$,按照递推公式(2-10),当 $r = n$ 时

$$b_0^{(n)}(t) = B(t) = \sum_{i=0}^{n} b_i B_{i,n}(t).$$

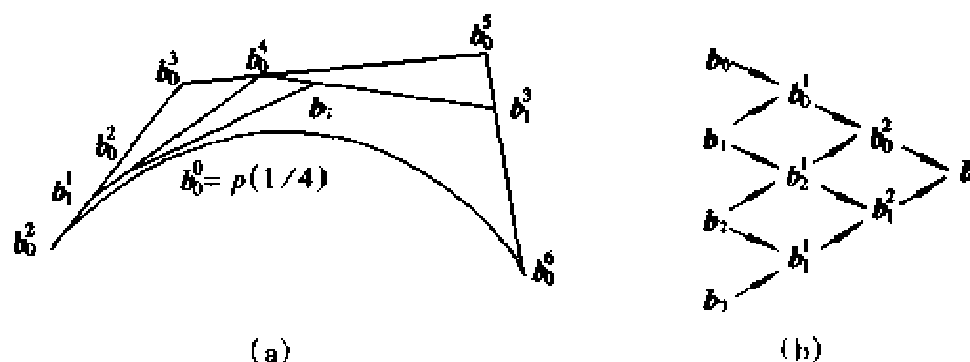


图 2-2

(a) 几何作图求贝齐尔曲线上一点($n = 3, t = 1/4$), (b) 求贝齐尔曲线上点的递推过程

2.2.3 贝齐尔曲线的导矢

贝齐尔曲线(2-9) 的一阶导矢可按下列公式计算:

$$p'(t) = \frac{\mathrm{d}p(t)}{\mathrm{d}t} = nq(t),$$

其中

$$q(t) = \frac{1}{n}p'(t) = \sum_{i=0}^{n-1} \Delta b_j B_{j,n-1}(t). \tag{2-11}$$

如图 2-3 所示,其中差分矢量 $\Delta b_j = b_{j+1} - b_j$ 控制多边形的边矢量 a_{j+i}. 把(2-11)式中的 n 条边矢量的起点置于原点,矢量的末端就成为控制顶点. 曲线 $q(t)$ 称为一阶速端曲线.

贝齐尔曲线的高阶导矢公式

$$p^{(k)}(t) = \frac{n!}{(n-k)!}\sum_{j=0}^{n-k} \Delta^k b_j B_{j,n-k}(t).$$

其中高阶向前差分矢量

$$\begin{aligned}\Delta^k b_j &= \Delta^{k-1} b_{j+1} - \Delta^{k-1} b_j \\ &= \sum_{i=0}^{k} (-1)^{k-i} C_k^i b_{j+1}.\end{aligned}$$

分别取 $t = 0, t = 1$,得 n 次贝齐尔曲线的首、末端点导矢

$$p^{(k)}(0) = \frac{n!}{(n-k)!}\Delta^k b_0,$$

$$p^{(k)}(1) = \frac{n!}{(n-k)!}\Delta^k b_{n-k}.$$

由德卡斯特里奥算法得

$$p^{(k)}(t) = \frac{n!}{(n-k)!}\Delta^k b_0^{n-k},$$

其中$\Delta^k b_0^{n-k}$表示中间顶点$b_0^{n-k}, b_1^{n-k}, \cdots, b_k^{n-k}$的$k$阶差分.

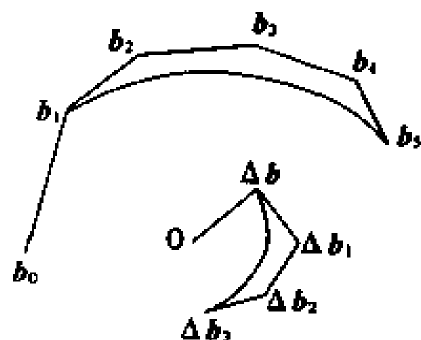


图 2-3

n 次贝齐尔曲线在首末端点处的曲率公式为

$$K(0) = \frac{n-1}{n}\frac{|\Delta b_0 \times \Delta b_1|}{|\Delta b_0|^3},$$

$$K(1) = \frac{n-1}{n}\frac{|\Delta b_{n-2} \times \Delta b_{n-1}|}{|\Delta b_{n-1}|^3}.$$

2.2.4　贝齐尔的升阶

在对贝齐尔曲线作修改时,有时通过增加控制点提高对曲线的灵活控制,而不改变原来曲线的形状. 为了实现此目的,对原有的贝齐尔曲线进行升阶是一种最简洁的方法. 如图 2-4 所示,原来由 4 个控制点 b_0, b_1, b_2, b_3 定义的曲线变为 5 个点 $b_0^{(1)}, b_1^{(1)}, b_2^{(1)}, b_3^{(1)}, b_4^{(1)}$,则贝齐尔曲线的表达式变为

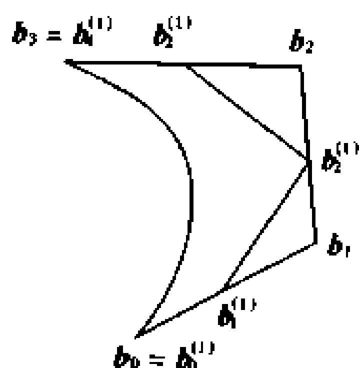


图 2-4

$$\sum_{j=0}^{n} C_n^j b_j t^j (1-t)^{n-j}$$
$$= \sum_{j=0}^{n+1} C_{n+1}^j b_j^{(1)} t^j (1-t)^{n+1-j}.$$

若对上式左边乘以$(t+(1-t))$,通过比较等式两边$t^j(1-t)^{n+1-j}$项的系数,得到

$$b_j^{(1)} = \frac{j}{n+1}b_{j-1} + \left(1 - \frac{j}{n+1}\right)b_j,$$
$$j = 0, 1, \cdots, n+1.$$

此式说明:

(1) 新的控制点 $b_j^{(1)}$ 是对原来特征多边形在参数$j/(n+1)$处进行线性插值的结果.

(2) 升阶后的新的特征多边形在原来特征多边形的凸包内.

(3) 升阶后的特征多边形更靠近贝齐尔曲线.

2.2.5　贝齐尔曲线的分割

给定控制顶点 $b_i, i = 0, 1, \cdots, n$ 及一参数值$t^* \in [0,1]$,由德卡斯特里奥算法求出所定义的贝齐尔曲线(2-9)上一点 $P(t^*)$,该点把曲线(2-9)分成两段:$p(t)$,

$t \in [0, t^*]$ 和 $p(t), t \in [t^*, 1]$，由图 2-2 知，顶点 $b_0, b_0^1, \cdots, b_0^n$ 为定义在 $t \in [0, t^*]$ 上那个子曲线段的贝齐尔点，顶点 $b_0^n, b_1^{n-1}, \cdots, b_n^0$ 为定义在 $t \in [t^*, 1]$ 上那个子曲线段的贝齐尔点．由这两组顶点可分别写出定义在各自局部参数域[0,1]上的两子曲线段的贝齐尔曲线表示形式(2-9)，这就是贝齐尔曲线在 t^* 的分割．

2.2.6　贝齐尔曲线间的连续拼接

1. 贝齐尔曲线间的参数连续性

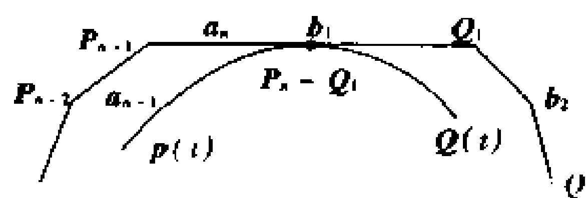


图 2-5

设给定两条贝齐尔曲线的控制点列 $P_i, i = 0, 1, \cdots, n$ 且 $a_i = P_i - P_{i-1}$ 和 Q_j，$j = 0, 1, \cdots, m$，且 $b_j = Q_j - Q_{j-1}$，如图 2-5 所示．设 $p(t)$、$q(t)$ 表示两条贝齐尔曲线，并且 $p(t)$ 的终点 P_n 和 $q(t)$ 的始点 Q_0 重合，即已达到 C^0 连续．

要使它们达到 C^1 连续的充要条件是，P_{n-1}、$P_n = Q_0$、Q_1 三点共线，即

$$b_1 = \alpha a_n(0).$$

要使它们达到 C^2 连续的充要条件是，在 C^1 连续的前提下再增加两个条件，即①密切平面重合，副法线矢量同向；②曲率相等．

由贝齐尔曲线的端点性质可知，$p(t)$ 在终点的副法线矢量是：

$$vp(1) = n^2(n-1)(a_{n-1} \times a_n);$$

$q(t)$ 在始点的副法线矢量

$$vq(0) = m^2(m-1)(b_1 \times b_2).$$

根据条件①必须导致四个矢量 a_{n-1}、a_n、b_1、b_2 共面，加上已有 $b_1 = \alpha a_n$ 的条件，则有：$b_2 = -\beta a_{n-1} + \gamma a_n, \beta > 0, \gamma$ 为任意数．根据 2.2.3 小节中的端点处曲率计算公式，从上述结论及条件②，可得到 $\beta = \dfrac{m(m-1)}{n(m-1)}\alpha^2$.

2. 贝齐尔曲线间的几何连续性

设 $B_1(t) = \sum_{i=0}^{n} b_i B_{i,n}(t), B_2(t) = \sum_{i=0}^{m} b_i^* B_{i,m}(t), t \in [0,1]$，分别为 n 次和 m 次的贝齐尔曲线．

(1) **C^0 拼接的充要条件**　由曲线几何光滑拼接条件易知，$B_1(t)$、$B_2(t)$ 是 C^0 拼接的充要条件是

$$B_1(1) = B_2(0)，即\ b_n = b_0^*.$$

(2) **G^1 光滑拼接的充要条件**　除上式外，还有 $B'_1(1) = aB'_2(0), a > 0$，即

$$n(b_n - b_{n-1}) = am(b_1^* - b_0^*),$$

b_{n-1}, b_n, b_1^* 共线．图 2-6 是 G' 光滑拼接图形．

(3) **G^2 光滑拼接的充要条件**　n 次贝齐尔曲线 $B_n(t)$ 与 m 次贝齐尔曲线 $B_2(t)$ G^2 光滑拼接的充要条件是

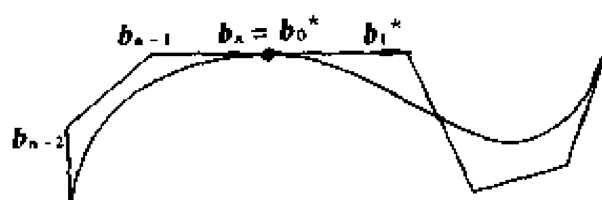


图 2-6

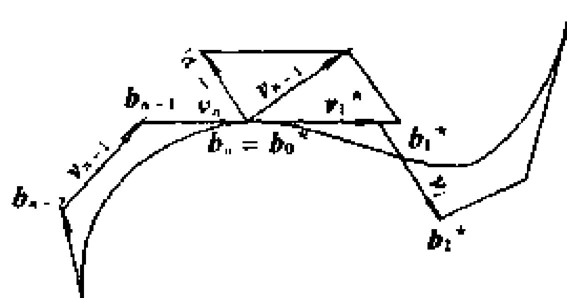


图 2-7

$$\begin{cases} b_n = b_0^*, \\ v_n = \dfrac{mav_1^*}{n}, \quad a > 0, \\ v_{n-1} = -\dfrac{m(m-1)}{n(n-1)} a^2 v_2^* + \eta v_1^*, \end{cases}$$

其中 $v_i = b_i - b_{i-1}, i = n-1, n; v_j^* = b_j^* - b_{j-1}^*, j = 1,2;$

$$\eta = a\frac{m}{n} + a^2\frac{m(m-1)}{n(n-1)} - \gamma\frac{m}{n(n-1)} \qquad (\gamma \text{ 为任意实数}),$$

$b_i(i = 0,1,\cdots,n)$、$b_j^*(j = 0,1,\cdots,m)$ 分别为 $B_1(t)$、$B_2(t)$ 的控制顶点. 图 2-7 是 G^2 光滑拼接图形.

2.2.7 有理贝齐尔曲线

对给定的控制顶点 $\{b_i\}_{i=0}^n$,称

$$R_n(t) = \frac{\sum_{i=0}^n w_i b_i B_{i,n}(t)}{\sum_{i=0}^n w_i B_{i,n}(t)} = \sum_{i=0}^n b_i R_{i,n}(t), \qquad t \in [0,1] \qquad (2\text{-}12)$$

为 **n 次有理贝齐尔曲线**,其中 $w_i \geqslant 0, i = 0,1,\cdots,n, \sum_{i=0}^n w_i = 1$,为有理贝齐尔曲线的权系数(若 $\sum_{i=0}^n w_i = w \neq 1$,可令 $w'_i = \frac{w^i}{w}$,使得 $\sum_{i=1}^n w'_i = 1$);$B_{i,n}$ 为 n 次的伯恩斯坦基函数,$R_{i,n}(t) = \dfrac{w_i B_{i,n}(t)}{\sum_{i=0}^n w_i B_{i,n}(t)}$ 为 n 次有理贝齐尔曲线的基函数,$i = 0,1,2,\cdots,n$.

(2-12) 式可看作是带权控制顶点 $(w_i b_i, w_i), i = 0,1,\cdots,n$,定义的非有理贝齐尔曲线在 $w = 1$ 超平面上的投影,若 $w_i = 1(i = 0,1,\cdots,n)$,得到非有理 n 次贝齐尔曲线.

引入权因子的作用是为了更好地控制曲线的形状. 当 $w_i > w_{i-1}$ 且 $w_i > w_{i+1}$ 时,就把曲线拉向 b_i 点. 如图 2-8 所示,$w_0 = w_1 = w_3 = 1$;当 $w_2 = 1/2,1,2$ 时,曲

线逐渐地靠近 b_2 点．

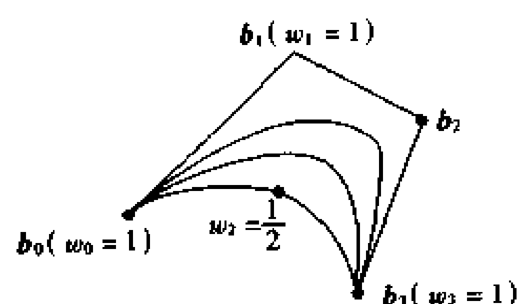


图 2-8

控制点 $b_i=(x_i,y_i,z_i)$ 可用齐次坐标表示成 $\{b_i^w=(w_ix_i,w_iy_i,w_iz_i,w_i)\}$，$i=0,1,\cdots,n$，因此(2-12)可用齐次坐标表示成

$$R_n(t)=H\left\{\sum_{i=0}^{n}b_i^wB_{i,n}(t)\right\}=\frac{\sum_{i=0}^{n}w_ib_iB_{i,n}(t)}{\sum_{i=0}^{n}w_iB_{i,n}(t)},$$

称点 $R_n(1/2)$ 为有理贝齐尔曲线的肩点．

对应有理基函数 $R_{i,n}(t)$ 的性质，有理贝齐尔曲线有下列性质：

(1) 凸包性．

(2) 几何不变性．

(3) 变差缩减性．

(4) 端点性质　$R_n(0)=b_0$，　$R_n(1)=b_n$，

$$R'_n(0)=\frac{nw_1}{w_0}(b_1-b_0),R'_n(1)=\frac{nw_{n-1}}{w_0}(b_n-b_{n-1}).$$

(5) 对于任意 $t\in[0,1]$，

$$\lim_{w_i\to+\infty}R_n(t)=\begin{cases}b_0, & t=0,\\ b_i, & t\in(0,1),\\ b_n, & t=1.\end{cases}$$

2.2.8　有理贝齐尔曲线的递推算法

对于 n 次有理贝齐尔曲线，可以建立类似于 n 次非有理贝齐尔曲线的递推算法．

设 $w_i^{(r)}(t)=\sum_{j=0}^{r}w_{i+j}B_{j,r}(t)$，则 $w_i^{(r-1)}(t)$、$w_i^{(r)}$ 和 $w_{i+1}^{(r-1)}(t)$ 之间有下面的关系式成立：

$$w_i^{(r)}(t)=(1-t)w_i^{(r-1)}+tw_{i+1}^{(r-1)}(t),\quad i=0,1,\cdots,n-r,\qquad(2\text{-}13)$$

若设中间点 $b_i^{(r)}(t)$：

$$b_i^{(r)}(t)=\frac{\sum_{j=0}^{r}w_{i+j}b_{i+j}B_{j,r}(t)}{w_i^{(r)}(t)}$$

则由(2-13)式可得中间点 $b_j^{(r)}(t)$、$b_i^{(r-1)}(t)$ 和 $b_{i+1}^{(r-1)}$ 之间有如下关系式成立：

$$b_i^{(r)}(t)=(1-t)\frac{w_i^{(r-1)}}{w_i^{(r)}(t)}b_i^{r-1}(t)+t\frac{w_{i+1}^{(r-1)}(t)}{w_i^{(r)}(t)}b_{i+1}^{(r-1)}(t),\quad i=0,1,\cdots,n-r,$$

于是 n 次有理贝齐尔曲线可写成下面的形式：

$$R(t)=b_0^n(t).$$

2.2.9 有理贝齐尔曲线的光滑拼接条件

1. 光滑拼接条件

与贝齐尔曲线一样，当控制点较多时，构造一条有理贝齐尔曲线通常是不方便的，实际上，一般的做法是构造若干个低次有理贝齐尔曲线，然后再根据光滑性要求，将它们光滑地拼接成一条样条曲线. 下面考虑两条有理贝齐尔曲线的光滑拼接条件.

设 $\boldsymbol{R}_n(t)=\dfrac{\sum_{i=0}^{n} w_i\boldsymbol{b}_iB_{i,n}(t)}{\sum_{i=0}^{n} w_iB_{i,n}(t)}$ 和 $\boldsymbol{R}_m(t)=\dfrac{\sum_{i=0}^{m} w_i^*\boldsymbol{b}_i^*B_{i,m}(t)}{\sum_{i=0}^{m} w_i^*B_{i,m}(t)}$ 是两条有理贝齐尔曲线，考虑它们在 $\boldsymbol{R}_n(1)$ 和 $\boldsymbol{R}_m(0)$ 处的光滑拼接条件.

(1) 由有理贝齐尔曲线的端点性质可知，$\boldsymbol{R}_m(0)$ 和 $\boldsymbol{R}_n(1)$ 相等的充要条件是

$$\boldsymbol{b}_n=\boldsymbol{b}_0^*. \tag{2-14}$$

(2) 为使 $\boldsymbol{R}_m(t)$ 与 $\boldsymbol{R}_n(t)$ 在 $\boldsymbol{R}_n(1)=\boldsymbol{R}_m(0)$ 处是 G^1 的，除了要具有上式外，还应有

$$\boldsymbol{R}'_n(0)=\alpha\boldsymbol{R}'_n(1).$$

将端点处切矢表示式代入上式得

$$\frac{n\alpha}{w_n}(\boldsymbol{b}_n-\boldsymbol{b}_{n-1})=\frac{m}{w_0^*}(\boldsymbol{b}_1^*-\boldsymbol{b}_0^*). \tag{2-15}$$

因此，要使 $\boldsymbol{R}_m(t)$ 与 $\boldsymbol{R}_n(t)$ 在连接处是 G^1 拼接的，充要条件是(2-14)和(2-15)式成立.(2-15)式表示 $\boldsymbol{b}_{n-1}$、$\boldsymbol{b}_n=\boldsymbol{b}_0^*$、$\boldsymbol{b}_1^*$ 在一条直线上.

(3) $\boldsymbol{R}_n(t)$ 与 $\boldsymbol{R}_m(t)$ 在 $\boldsymbol{R}_m(0)=\boldsymbol{R}_n(1)$ 处是 G^2 的充要条件是，

1° 它们是 G^1 连续的；

2° 副法向量

$$\boldsymbol{B}_n(1)=\frac{n^2(n-1)w_{n-1}w_{n-2}}{w_n^2}[(\boldsymbol{b}_{n-1}-\boldsymbol{b}_{n-2})\times(\boldsymbol{b}_n-\boldsymbol{b}_{n-1})]$$

与

$$\boldsymbol{B}_m(0)=\frac{m^2(m-1)w_1^*w_2^*}{w_0^{*2}}[(\boldsymbol{b}_1^*-\boldsymbol{b}_0^*)\times(\boldsymbol{b}_2^*-\boldsymbol{b}_1^*)]$$

有相同的方向；

3° $\boldsymbol{R}_n(t)$ 和 $\boldsymbol{R}_m(t)$ 在 $\boldsymbol{R}_m(0)=\boldsymbol{R}_n(1)$ 处的曲率相同.

由 1° 和 2° 可知，$\boldsymbol{b}_{n-2}$、$\boldsymbol{b}_{n-1}$、$\boldsymbol{b}_n=\boldsymbol{b}_0^*$、$\boldsymbol{b}_1^*$、$\boldsymbol{b}_2^*$ 是共面的，加上条件(2-15)，则有

$$\boldsymbol{b}_2^*-\boldsymbol{b}_1^*=-\beta(\boldsymbol{b}_{n-1}-\boldsymbol{b}_{n-2})+\eta(\boldsymbol{b}_n-\boldsymbol{b}_{n-1}),$$

其中 $\beta>0$，η 为任意常数. 在 $\boldsymbol{R}_m(0)=\boldsymbol{R}_n(1)$ 处两曲线的曲率分别为

$$\begin{cases} K_n(1) = \dfrac{(n-1)w_n w_{n-2}}{n w_{n-1}^2} \dfrac{|(b_{n-1}-b_{n-2})\times(b_n-b_{n-1})|}{|b_n-b_{n-1}|^3}, \\ K_m(0) = \dfrac{(m-1)w_0^* w_2^*}{m w_1^{*2}} \dfrac{|(b_1^*-b_2^*)\times(b_0^*-b_1^*)|}{|b_0^*-b_1^*|^3}. \end{cases}$$

可把 $K_m(0)$ 表示成

$$K_m(0) = \frac{-m(m-1)w_2^* w_n^2 \beta}{w_0^*(w_1^*)^3 a^2} \frac{|(b_{n-1}-b_{n-2})\times(b_n-b_{n-1})|}{|b_{n-1}-b_n|^3},$$

由条件 3°: $K_m(0) = K_n(1)$, 得

$$\beta = \frac{n(n-1)w_{n-2}w_0^*(w_1^*)^2 a^2}{m(m-1)w_n w_2^* w_{n-1}^2}.$$

上面所用的 $\boldsymbol{B}_m(0)$、$\boldsymbol{B}_n(0)$ 及 $K_m(1)$ 分别表示 $\boldsymbol{R}_m(t)$、$\boldsymbol{R}_n(t)$ 在 $t=0$ 及 $t=1$ 处的副法向量和曲率.

2.3 B样条曲线、曲面

2.3.1 B样条曲线的定义和性质

B样条曲线方程可写为

$$C(u) = \sum_{i=0}^{n} P_i N_{i,k}(u), \qquad u \in [a,b], \tag{2-16}$$

其中 P_i, $i=0,1,\cdots,n$, 为控制顶点的矢径, P_i 又称 de Boor 点. 顺序连成的折线称为B样条控制多边形, $N_{i,k}(u)$, $i=0,1,\cdots,n$ 称为 k 次B样条基函数, 当 $a=0$, $b=1$ 时, B样条基函数称为是规范的. 它由称为节点矢量的参数 u 的非减序列 $a=u_0 \leqslant u_1 \leqslant \cdots \leqslant u_{n+k+1}=b$ 所决定.

B样条曲线与贝齐尔曲线比较, 差别在于:

(1) 对于贝齐尔曲线, 基函数的次数等于控制顶点数减1. 对于B样条曲线, 基函数的次数 k 与控制顶点数无关.

(2) 贝齐尔曲线的基函数即伯恩斯坦基函数是多项式函数. B样条曲线的基函数即B样条基函数是多项式样条.

(3) 贝齐尔曲线是一种特殊表示形式的参数多项式曲线. B样条曲线则是一种特殊表示形式的参数样条曲线.

(4) 贝齐尔曲线缺乏局部性质, B样条曲线具有局部性质. 局部性质是B样条曲线的最重要的性质之一.

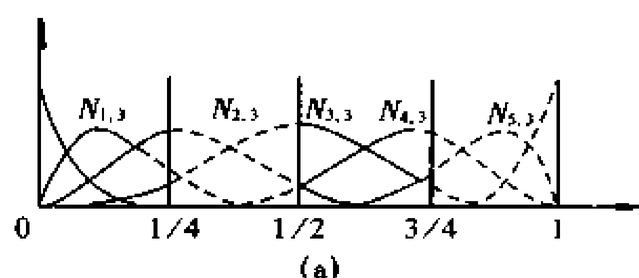
$n+1$ 个控制顶点的 k 次B样条曲线的定义域为 $u \in [u_k, u_{n+1}]$, 共含有 $n-k+1$ 个节点区间(包括零长度的节点区间), 若其中不含重节点, 则B样条曲线包含 $n-k+1$ 段. k 次B样条曲线方程(2-16)可改写成为分段表示:

$$C(u) = \sum_{j=i-k}^{i} P_j N_{j,k}(u), \qquad u \in [u_i, u_{i+1}]. \tag{2-17}$$

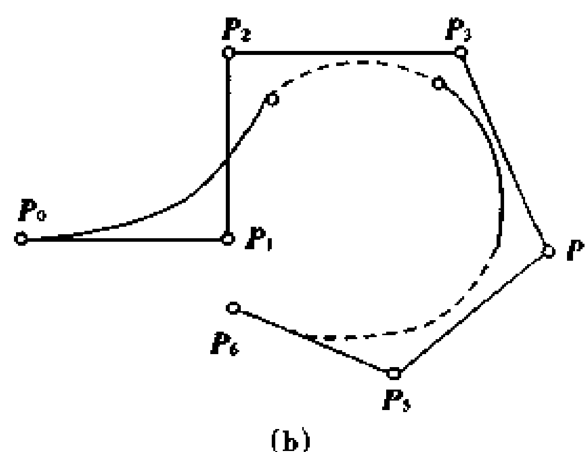
利用 B 样条基函数 $N_{i,k}(u)$ 的性质，B 样条曲线有下列性质：

(1) 若 $n = k$，且节点矢量 $U = (0,\cdots,0,1,\cdots,1)$，则 $C(u)$ 是贝齐尔曲线.

(2) $C(u)$ 是分段多项式曲线，次数 k、控制顶点数 $n+1$ 和节点数 $m+1$ 有下列关系（考虑重节点情形）：$m = n + k + 1$. 如图 2-9 所示.



(a)



(b)

图 2-9

(3) 端点性质　$C(0) = C_0, C(1) = c_n$.

(4) 仿射不变性.

(4) 强凸包性质　有着比贝齐尔曲线更强的凸包性质.

(5) 局部性质　k 次 B 样条曲线上参数为 $u \in [u_i, u_{i+1}]$ 的点与 $k+1$ 个控制顶点 $P_j(j = i-k,\cdots,i)$ 有关，与其它控制顶点无关（考虑重节点情形）；移动该曲线的第 i 个控制顶点 P_i 只影响到定义在 $[u_i, u_{i+k+1}]$ 上那部分曲线的形状，对曲线的其余部分不发生影响.

(6) 控制多边形表示对曲线的分段线性逼近. 在相同的控制多边形和相同的节点矢量下，曲线次数越低，曲线越靠近控制多边形. 因此 B 样条曲线具有磨光性质. B 样条曲线随着次数升高，越来越光滑，如图 2-10.

(7) 曲线从 $u = u_0$ 到 $u = u_{n+k+1}$，$N_{i,k}(u)$ 的作用如一个开关，当 u 通过一个节点时，一个基函数不起作用（$= 0$），下一个基函数起作用（$\neq 0$）.

(8) 可微性或参数连续性　B 样条曲线在每一曲线段内部是无限次可微的（C^{∞}），在对应节点的曲线段端点处是 $k - r$ 次可微（C^{k-r}），r 是该节点的重复度.

(9) 变差缩减性质（与贝齐尔曲线类似）.

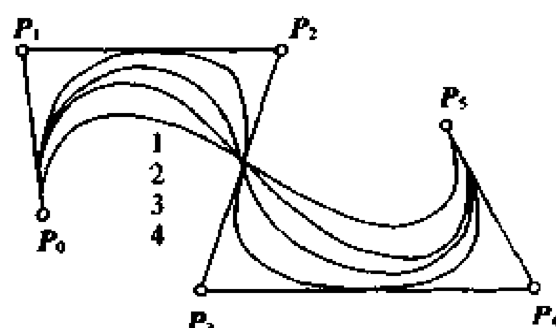


图 2-10

2.3.2 重节点与重顶点对B样条曲线的影响

1. 重节点的影响

节点矢量 U 的节点分布决定B样条基的几何形状和数目，而B样条基函数的作用，相当于对控制多边形的磨光.

(1) 在B样条曲线定义域内的内重节点，重复度每增加1，曲线段数减1，样条曲线在该重节点处的可微性或参数连续阶降1. 因此，k 次B样条曲线在重复度为 r 的节点处是 C^{k-r} 连续的. 一条位置连续的曲线，其内节点所取的最大重复度等于曲线的次数 k，端节点的最大重复度为 $k+1$. 依据这一性质，可以在B样条曲线内部构造尖点与尖角(注意与重顶点构造尖角的区别). 甚至两条或多条分离的B样条曲线可以采用一个统一的方程表示. 如图 2-11.

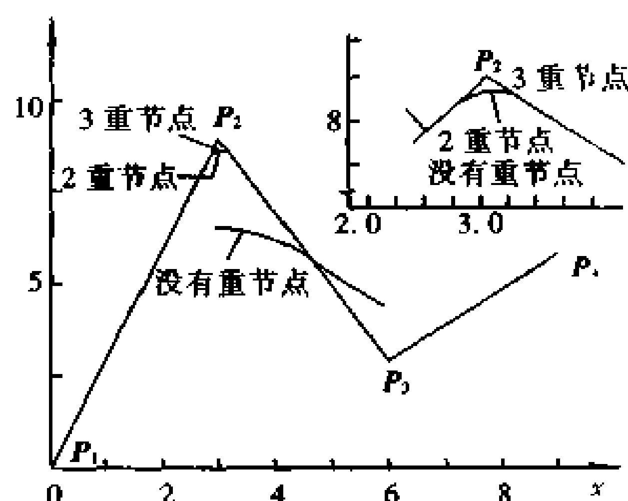


图 2-11

(2) 当端节点重复度为 k 时，k 次B样条曲线的端点将与相应的控制多边形的端顶点相重，并在端点处与控制多边形相切.

(3) 当在曲线定义域内有重复度为 k 的节点时，k 次B样条曲线插值于相应的控制顶点. 与设置 k 重顶点达到插值顶点不同之处在于，不致引起曲线在该点处切

矢消失，保持曲线的正则性.

(4) 当端节点重复度为 $k+1$ 时，k 次 B 样条曲线就具有和 k 次贝齐尔曲线相同的端点几何性质.

(5) k 次 B 样条曲线若在定义域内相邻两节点都具有重复度 k，可以生成定义在该节点区间上那段 B 样条曲线的贝齐尔点.

(6) 若端节点重复度为 $k+1$ 的 k 次 B 样条曲线的定义域仅有一个非零节点区间，则所定义的该 k 次 B 样条曲线就是 k 次贝齐尔曲线.

2. 重顶点的影响

当顺序 $k+1$ 个顶点共线时，由该 $k+1$ 个顶点定义的 k 次 B 样条曲线段为一直线. 当顺序 $k+1$ 个顶点相重时，所定义的那一曲线段就退化为那个重合点，而前后邻段又因 $k+1$ 个顶点共线而形成两直线，从而在退化处形成尖角，如图 2-12 所示. 对于参数曲线而言，在重顶点处，曲线是 C^{k-1} 连续的，但曲线在该点处是奇点 $k+1$ 重，该点处切矢间断. 由此可见，在非正则点处，参数连续性与曲线的光滑度出现不一致. 重节点、重顶点都可以生成曲线上的尖点.

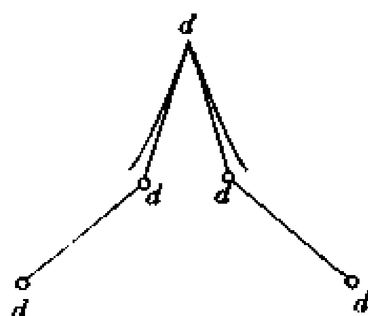


图 2-12

2.3.3 B 样条曲线的类型划分

设给定控制顶点 $\boldsymbol{P}_i$，$i=0,1,\cdots,n$，次数为 k，节点矢量为 $\boldsymbol{U}=(u_0,u_1,\cdots,u_{n+k+1})$. B 样条曲线按节点矢量中节点的分布，可分为三种类型：

(1) 均匀 B 样条曲线　节点矢量各节点之间间距相等，即 $\Delta u_i=u_{i+1}-u_i=$ 常数.

(2) 准均匀 B 样条曲线　首端和末端节点均为 $k+1$ 重节点，内节点间等距.

(3) 非均匀 B 样条曲线　节点矢量只要求为非减序列. 通常两端节点取 $k+1$ 重点，使曲线两端具有贝齐尔曲线的端点性质.

另外，B 样条曲线可以按端点连接情形分为周期曲线与非周期曲线.

2.3.4 B 样条曲线的分割和节点插入算法

1. de Boor 分割算法

k 次 B 样条基函数可由两个相邻的 $k-1$ 次 B 样条基函数线性组合而成，利用 k 次 B 样条函数的性质，可用 de Boor 算法计算曲线段 $\boldsymbol{C}(u)$ 的值. 若 $u\in[u_j,u_{j+1}]$，$k\leqslant j\leqslant n$，$\boldsymbol{C}(u)=\boldsymbol{P}_j^{k-1}(u)$，其递推公式是：

$$\boldsymbol{C}_i^{(r)}(u)=\begin{cases}\boldsymbol{P}_i, & r=0,\\ \dfrac{u-u_i}{u_{i+k+1-r}-u_i}\boldsymbol{C}_i^{(r-1)}(u)+\dfrac{u_{i+k+1-r}-u}{u_{i+k+1-r}-u_j}\boldsymbol{C}_{i-1}^{(r-1)}(u), & \\ & r=1,2,\cdots,k,\ i=j-k,\cdots,j.\end{cases}$$

上式亦可表示成

$$C(u)=\sum_{i=j-k+1}^{j}P_iN_{i,k}(u)=\sum_{i=j-k+2}^{j}P_i^1N_{i,k-1}(u)=\cdots=P_j^{k-1}(u).$$

求曲线上一点的递推过程可以表示成下述的三角形,其几何意义如图 2-13 所示.

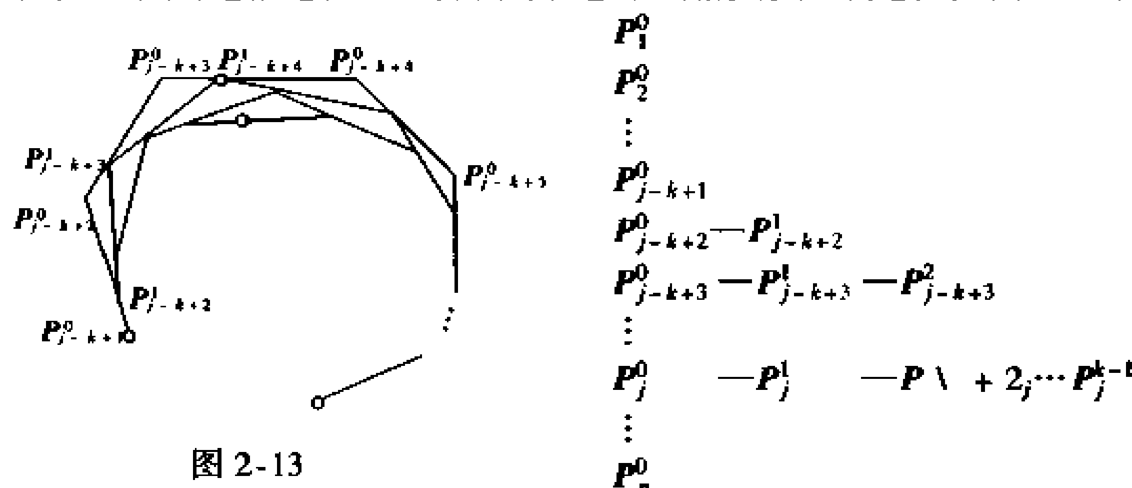


图 2-13

de Boor 分割算法的实质是用 $P_i^rP_{i+1}^r$ 构成的边切 P_i^{r-1} 角,从图 2-13 可见,从多边形 $P_{j-k+1}P_{j-k+2}\cdots P_j$ 开始经过 $(k-1)$ 层的切角,最后得到 $k+1$ 层上的点 $P_j^{k-1}(u)$.图 2-13 为 B 样条曲线 de Boor 算法的几何意义.

2. Oslo 节点插入算法

假设 k 次 B 样条曲线的节点矢量 $U=(u_0,u_1,\cdots,u_{n+k+1})$,且为非减序列;控制点序列为 $P_0,P_1,\cdots,P_n$,在节点区间 $[u_i,u_{i+1}]$ 中 u 处插入 r 次 $(r+s<k)$,s 表示 u 处原节点重数,控制点中与区间 (u_i,u_{i+1}) 相关的 $k+1$ 个原始控制点为:P_{i-k},$P_{i-k+1},\cdots,P_{i-1},P_i$.经过节点插入后,曲线不变,新的控制点为

$$Q_{j,r}=\alpha_{j,r}Q_{j,r-1}+(1-\alpha_{j,r})Q_{j-1,r-1},\tag{2-18}$$

其中

$$\alpha_{j,r}=\begin{cases}1, & j<i-k+r-1,\\ \dfrac{u-u_i}{u_{j+k-r+1}-u_i}, & i-k+r\leqslant j\leqslant i-s,\\ 0, & j\geqslant i-s+1.\end{cases}$$

2.3.5 反求 B 样条曲线的控制点及其端点性质

所谓反求 B 样条曲线控制点,是指已知一组空间型值点 $Q_i(i=1,2,\cdots,n)$,要找一条 k 次(这里以常用的三次为例)B 样条曲线 $C_j(u)$ 过 Q_i 点,亦即找一组与点列 Q_i 对应的 B 样条特征多边形顶点 $P_j(j=0,1,\cdots,n+1)$.对于均匀的三次 B 样条曲线,其上的型值点和控制点的位置矢量之间有关系:

$$P_j(0)=(P_{j-1}+4P_j+P_{j+1})/6=Q_j,\quad j=1,2,\cdots,n.\tag{2-19}$$

(2-19) 式有 n 个方程,但有 $n+2$ 个未知数,需要补充两个边界条件:

1. 首末两点过 Q_1 和 Q_n 的非周期三次 B 样条曲线

此时应有 $P_1=Q_1,P_n=Q_n$,于是求解控制点 P_j 的线性方程组为

$$\begin{bmatrix} 6 & & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & & \ddots & & \\ & & & 1 & 4 & 1 \\ & & & & & 6 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = 6 \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{n-1} \\ Q_n \end{bmatrix}$$

这是一个主对角线占优的三带状矩阵，故可用追赶法解出 $P_j(j = 1,2,\cdots,n)$. 为了使曲线的首末两点过 Q_1 和 Q_n，需要两个附加顶点 P_0、P_{n+1}，且应满足条件 $P_0 = 2P_1 - P_2$, $P_{n+1} = 2P_n - P_{n-1}$，在此情况下所生成的 B 样条曲线两端点处的曲率为零，即曲线首末端分别与 P_0P_1 及 P_nP_{n-1} 相切.

2. 封闭的三次 B 样条曲线

为保证曲线能首尾相接，并使曲线上结点序号与特征多边形顶点序号相对应，即有 $P_0 = P_n$, $P_{n+1} = P_1$，于是可得到线性方程组：

$$\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & & \ddots & & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_{n-1} \\ P_n \end{bmatrix} = 6 \begin{bmatrix} Q_1 \\ Q_2 \\ Q_{n-1} \\ Q_n \end{bmatrix}.$$

3. 端点有二重控制点的三次 B 样条曲线

此时应有：$P_0 = P_1$, $P_{n+1} = P_n$，由此可构成线性方程组：

$$\begin{bmatrix} 6 & -6 & & & \\ 1 & 4 & 1 & & \\ & & \ddots & & \\ & & 1 & 4 & 1 \\ & & & 6 & -6 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = 6 \begin{bmatrix} 0 \\ Q_1 \\ \vdots \\ Q_n \\ 0 \end{bmatrix}$$

由 $P_j(j = 0,1,2,\cdots,n,n+1)$ 控制点构造的三次 B样条曲线过 $Q_i(i = 1,2,\cdots,n)$.

4. 给定始、终点的切矢量 Q'_1、Q'_n

在始点，由于 $Q'_i = (P_2 - P_0)/2$, $Q_1 = (P_0 + 4P_1 + P_2)/6$，有 $P_0 = P_2 - 2Q'_1$, $2P_1/3 + P_2/3 = Q_1 + Q'_1/3$;

在终点，由于 $Q'_i = (P_{n+1} - P_0)/2$, $Q_n = (P_{n-1} + 4P_n + P_{n+1})/6$，有 $P_{n-1} = P_{n+1} - 2Q'_n$, $P_{n-1}/3 + 2P_n/3 = Q_n - Q'_n/3$.

把上述结果写成线性方程组：

$$\begin{bmatrix} 4 & 2 & & & \\ 1 & 4 & 1 & & \\ & & \ddots & & \\ & & 1 & 4 & 1 \\ & & & 2 & 4 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = 6 \begin{bmatrix} Q_1 + Q'_1/3 \\ Q_2 \\ \vdots \\ Q_{n-1} \\ Q_n - Q'_n/3 \end{bmatrix},$$

解得 $P_j(j = 1,2,\cdots,n)$ 后即可方便地求得 P_0 和 P_{n+1}.

由 $P_j(j=0,1,\cdots,n+1)$ 控制点生成的三次B样条曲线即满足端点切矢量为 Q'_1 和 Q'_n 的条件.

5. 给定始、终点的二阶导数矢量 R_1 和 R_n

在始点处,由于 $R_1=P_2-2P_1+P_0$,和 $R_1=6Q_1-6P_1$ 有 $P_0=2P_1-P_2+R_1$, $P_1=Q_1-R_1/6$.

在终点处,同样有 $P_{n+1}=2P_n-P_{n-1}+R_n$, $P_n=Q_n-R_n/6$.

由上述条件构造的线性方程组是

$$\begin{bmatrix}6 & & & & & \\ 1 & 4 & 1 & & & \\ & & \ddots & & & \\ & & & 1 & 4 & 1 \\ & & & & & 6\end{bmatrix}\begin{bmatrix}P_1\\ P_2\\ \vdots\\ P_{n-1}\\ P_n\end{bmatrix}=6\begin{bmatrix}Q_1+R_1/6\\ Q_2\\ \vdots\\ Q_{n-1}\\ Q_n-R_n/6\end{bmatrix}.$$

求解此方程组得到 $P_j(j=1,2,\cdots,n)$,再用上述条件即可求得 P_0 和 P_{n+1},由控制点列构造的三次B样条曲线过 $Q_i(i=1,2,\cdots,n)$,并满足端点处以 R_1 和 R_n 为二阶导数矢量的条件.

2.3.6　B样条曲线的导矢

设B样条曲线表示为

$$C(u)=\sum_{i=0}^{n}N_{i,p}(u)P_i,$$

节点矢量 $U=(\underbrace{0,\cdots,0}_{p+1},\cdots,u_{p+1},\cdots,u_{m-p-1},\underbrace{1,\cdots,1}_{p+1})$. 由于B样条曲线的导矢 $C'(u)$ 也表示一条B样条曲线,利用 de Boor 算法的推递公式,可写出导矢公式如下:记

$$C(u)=C^{(0)}(u)=\sum_{i=0}^{n}N_{i,p}(u)P_i^{(0)},$$

则

$$C^{(k)}(u)=\sum_{i=0}^{n-k}N_{i,p-k}(u)P_i^{(k)},$$

式中

$$P_i^{(k)}=\begin{cases}P_i, & k=0\\ \dfrac{p-k+1}{u_{i+p+1}-u_{i+k}}(P_{i+1}^{(k-1)}-P_i^{(k-1)}), & k>0,\end{cases}$$

$$U^{(k)}=(\underbrace{0,0,\cdots,0}_{p-k+1},u_{p+1},\cdots,u_{m-p-1},\underbrace{1,\cdots,1}_{p-k+1}).$$

可见,p 次B样条的 k 阶导矢可表示成 $p-k$ 次B样条曲线. 其中,曲线端点处的切矢有:

$$C'(0)=\frac{p}{u_{p+1}}(P_1-P_0),$$

$$C'(1)=\frac{p}{1-u_{m-p-1}}(P_n-P_{n-1}).$$

2.3.7 张量积 B 样条曲面

1. 定义和性质

给定 $(n+1)\times(m+1)$ 个控制顶点 P_{ij}，$i=0,1,\cdots,n$，$j=0,1,\cdots,m$，构成一张控制网格，又设节点矢量(取标准型)为

$$\boldsymbol{U}=(\underbrace{0,\cdots,0}_{p+1},u_{p+1},\cdots,u_{r-p-1},\underbrace{1,\cdots,1}_{p+1}),\boldsymbol{V}=(\underbrace{0,0,\cdots,0}_{q+1},v_{q+1},\cdots,v_{s-q-1},\underbrace{1,\cdots,1}_{q+1}),$$

则 $p\times q$ 次张量积 B 样条曲面定义如下：

$$\boldsymbol{S}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)\boldsymbol{P}_{ij},\tag{2-20}$$

其中节点矢量 $\boldsymbol{U}$ 有 $r+1$ 个节点，$\boldsymbol{V}$ 有 $s+1$ 个节点，并且有

$$r=n+p+1 \text{ 和 } s=m+q+1.$$

B 样条曲面可以写成如下矩阵形式：

$$\boldsymbol{S}(u,v)=[N_{k,p}(u)]^{\mathrm{T}}[\boldsymbol{P}_{kl}][N_{l,q}(u)],\quad i-p\leqslant k\leqslant i,\quad j-q\leqslant l\leqslant j\tag{2-21}$$

式中$[N_{k,p}(u)]^{\mathrm{T}}$是 $1\times(p+1)$ 行向量，$[\boldsymbol{P}_{kl}]$ 是$(p+1)\times(q+1)$ 阶控制点阵，$[N_{1,q}(v)]$ 是$(q+1)\times 1$ 列向量.

对应双变量张量积 B 样条基函数的性质，张量积 B 样条曲面有下列性质：

(1) 若 $n=p$，$m=q$，$\boldsymbol{U}=(0,\cdots,0,1,\cdots,1)$ 和 $\boldsymbol{V}=(0,\cdots,0,1,\cdots,1)$，则 $\boldsymbol{S}(u,v)$ 退化为贝齐尔曲面；

(2) 角点插值，$\boldsymbol{S}(0,0)=\boldsymbol{P}_{0,0}$，$\boldsymbol{S}(1,0)=\boldsymbol{P}_{n,0}$，$\boldsymbol{S}(0,1)=\boldsymbol{P}_{0,m}$，$\boldsymbol{S}(1,1)=\boldsymbol{P}_{n,m}$；节点矢量 $\boldsymbol{U}$、$\boldsymbol{V}$ 两端点取 p，q 重节点，保证角点插值；

(3) 仿射不变性；

(4) 强凸包性；

(5) 控制网格是对曲面的分片平面片逼近，与曲线情形类似，曲面次数越低，这样的逼近性越好；

(6) 局部性质，P_{ij} 只影响矩形域$[u_i,u_{i+p+1}]\times[V_j,V_{j+q+1}]$上的曲面形状；

(7) 可微性：$\boldsymbol{S}(u,v)$ 是沿 u 向(或 v 向) 在重数是 k 的节点 u(或 v) 处，是 p-k(或 q-k) 阶可微的.

以上性质都是 B 样条曲线相应性质的直接推广，但是对曲面而言，没有变差缩减性质.

2. B 样条曲面的导数

考察(2-20) 式的偏导数

$$\frac{\partial^{k+l}}{\partial u^k\partial v^l}\boldsymbol{S}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}^{(k)}N_{j,q}^{(l)}\boldsymbol{P}_{ij},$$

利用 B 样条曲线的求导公式，有

$$\boldsymbol{S}_u(u,v)=\sum_{i=0}^{n-1}\sum_{j=0}^{m}N_{i,p-1}(u)N_{j,q}(v)\boldsymbol{P}_{ij}^{(1,0)},$$

其中

$$P_{ij}^{(1,0)} = p\frac{P_{i+1,j} - P_{ij}}{u_{i+p+1} - u_{i+1}},$$

$$U^{(1)} = (\underbrace{0,0,\cdots,0}_{p}, u_{p+1},\cdots,u_{r-p-1},\underbrace{1,\cdots,1}_{p}),$$

$$V^{(0)} = V.$$

类似也有

$$S_v(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m-1} N_{i,p}(u)N_{j,q-1}(v)P_{ij}^{(0,1)},$$

其中

$$P_{ij}^{(0,1)} = q\frac{P_{i,j+1} - P_{ij}}{v_{j+q+1} - v_{j+1}},$$

$$U^{(0)} = U,$$

$$V^{(1)} = (\underbrace{0,\cdots,0}_{q}, v_{q+1},\cdots,v_{s-q-1},\underbrace{1,\cdots,1}_{q}).$$

从而

$$S_{uv}(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} N_{i,p-1}(u)N_{j,q-1}(v)P_{ij}^{(1,1)},$$

其中 $P_{ij}^{(1,1)} = q\frac{P_{i,j+2}^{(1,0)} - P_{ij}^{(1,0)}}{v_{j+q+1} - v_{j+1}}$，$U^{(1)}$，$V^{(1)}$ 如上所定义. 一般地有

$$\frac{\partial^{k+l}}{\partial u^k \partial v^l}S(u,v) = \sum_{i=0}^{n-k}\sum_{j=0}^{m-l} N_{i,p-k}(u)N_{j,q-l}(v)P_{ij}^{(k,l)},$$

其中

$$P_{ij}^{(k,l)} = (q-l+1)\frac{P_{i,j+1}^{(k,l-1)} - P_{ij}^{(k,l-1)}}{v_{j+q+1} - v_{j+1}}.$$

3. B 样条曲面的反算

B 样条曲面的反算或逆过程就是要构造一张 $p \times q$ 次 B 样条曲面插值. 给定呈拓扑矩形阵列的数据点 $d_{i,j}$，$i = 0,1,\cdots,m$；$j = 0,1,\cdots,n$，数据点阵中每一排数据点都位于曲面的一条等参数线上. 曲面反算问题虽然也能像曲线反算那样，表达为求解未知控制顶点 $p_{i,j}$；$i = 0,1,\cdots,m+p-1$；$j = 0,1,\cdots,n+q-1$ 的一个线性方程组，但当 m、n 较大时，计算较为复杂，一种方法是表达为张量积曲面计算的逆过程，它把曲面的反算问题化解为两阶段的曲线反算问题. 设待求的 B 样条插值曲面方程可写成为

$$S(u,v) = \sum_{i=0}^{m+p-1}\sum_{j=0}^{n+q-1} N_{i,p}(u)N_{j,q}(v)P_{ij},$$

反算曲面控制顶点 P_{ij} 的算法如下：

(1) 数据点阵的参数化　一种方法是对一个方向的每一排数据取平均值，再用曲线的数据参数化方法（如累加弦长法），对取平均值后的数据值参数化. 设 u 向参数化为 $\{t_i\}_{i=0}^{m}$. 同样可对 v 向数据作参数化：$\{s_j\}_{j=0}^{n}$.

(2) 求 u、v 向的节点矢量 U、V　$u_0 = \cdots = u_p = 0$，$u_{m-p} = \cdots = u_m = 1$，$u_{i+p} = \frac{1}{p}\sum_{j=i}^{i+p-1} t_j$，　$i = 1,\cdots,m-p$，

$$U = (\underbrace{0,\cdots,0}_{p+1}, u_{p+1},\cdots,u_m,\underbrace{1,\cdots,1}_{p+1}).$$

同理有 $V = (\underbrace{0,\cdots,0}_{q+1}, v_{q+1},\cdots,v_n,\underbrace{1,\cdots,1}_{q+1})$;

(3) 对 u 向 $m+1$ 排数据 $\{d_{ij}\}_{j=0}^{n}, i = 0,\cdots,m$,分别作 q 次 B 样条曲线插值生成曲线 $C_i(v)$,反算各排曲线的控制顶点 $\{R_{ij}\}_{j=0}^{n+q-1}, i = 0,\cdots,m$.

以 U 为带点矢量,对每一 $j = 0,\cdots,n+q-1$,对 $m+1$ 个数据点 $\{R_{ij}\}_{i=0}^{m}$ 作 p 次 B 样条曲线插值,反算出 $n+q$ 排控制顶点 $\{P_{ij}\}, i = 0,\cdots,m+p-1, j = 0,\cdots,n+q-1$.此控制点阵,即为所求的 $p \times q$ 次 B 样条曲面的控制顶点点阵.

2.4 NURBS 曲线、曲面

2.4.1 NURBS 曲线

曲线、曲面采用 NURBS 表示,有下列优点:

(1) 对标准的曲面(如圆锥曲面、二次曲面、回转面等)和自由曲线、曲面提供了统一的数学表示,便于工程数据库的存取和应用.

(2) 可通过控制点和权因子来灵活地改变形状.

(3) 对插入节点、修改、分割、几何插值等的处理工具比较有力.

(4) 具有透视投影变换和仿射变换的不变性.

(5) 非有理 B 样条、有理及非有理贝齐尔曲线、曲面是 NURBS 的特例表示.

但是,目前应用 NURBS 中还有一些难以解决的问题:

(1) 比一般的曲线、曲面定义方法更费存储空间和处理时间.

(2) 权因子选择不当会造成形状畸变.

(3) 对搭接、重叠形状的处理相当麻烦.

(4) 像点的映射这类算法在 NURBS 情况下会变得不太稳定.

1. 定义和性质

(1) NURBS 曲线的定义

NURBS 曲线是由 B 样条多项式基函数定义的,形式是

$$C(u) = \frac{\sum_{i=0}^{n} w_i P_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)} = \sum_{i=0}^{n} P_i R_{i,k}(u) \tag{2-22}$$

其中 P_i 是特征多边形顶点位置矢量,$N_{i,k}(u)$ 是由节点矢量

$$U = (\underbrace{a,\cdots,a}_{k+1}, u_{k+1},\cdots,u_n,\underbrace{b,\cdots,b}_{k+1})$$

决定的 k 次 B样条基函数,w_i 是相应控制点 P_i 的权因子,节点向量中节点个数 $m = n+k+1$,n 为控制点数,k 为 B 样条基函数的次数.$R_{i,k}(u)$ 是有理 B 样条基函数.

在实际应用中常取 $a = 0, b = 1$.由(2-22)式和节点矢量 U 定义的 $u \in [0,1]$ 区间上的整条 NURBS 曲线与贝齐尔曲线相似,即曲线过起、终点,且起、终点的切

矢量是控制多边形的第一条边和最后一条边.

(2) 在齐次坐标下 NURBS 的几何意义

先考虑平面NURBS曲线的情况,非平面NURBS曲线和曲面可看成是这种情况下的推广,如图 2-14 所示,在 $oxyw$ 坐标系中的每一个点,若 $w \neq 0$,可表示成(xw, yw, w),$(x, y, 0)$ 表示 oxy 平面上的无穷远点,这些点经透视变换映射到 xy 平面后是:

图 2-14

$$\varphi(x, y, w) = \begin{cases} \left(\dfrac{x}{w}, \dfrac{y}{w}\right), & w \neq 0, \\ \text{取}(x, y)\text{的方向}, & w = 0. \end{cases}$$

如已知具有权因子的控制点集,构造相应的 NURBS 曲线可采取下述步骤:

1) 构造具有权因子的顶点 $P_i^w = (w_i x_i, w_i y_i, w_i)$,　$i = 0, 1, \cdots, n$;

2) 在 o-xyw 坐标系下得到一条非有理 B 样条曲线 $C^w(u) = \sum_{i=0}^{n} P_i^w N_{i,k}(u)$;

3) 把 $C^w(u)$ 映射到 $w = 1$ 平面上,$C(u) = \varphi(C^w(u)) = \dfrac{\sum_{i=0}^{n} w_i P_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)}$.

上述讨论中给出了在投影平面上 NURBS 的几何定义,$C^w(u)$ 是在三维欧氏空间中定义的 NURBS 曲线,而 $C(u)$ 是在二维欧氏空间的 NURBS 曲线,NURBS 曲线的三维模型的建立不仅对显示曲线有帮助,并且对构造 NURBS 曲线生成算法也是必不可少的.

(3) 仿射、透视变换的不变性

1° 仿射变换　对一个点 P 作仿射变换,即为 $A[P] = L[P] + T$.可以证明 NURBS 曲线 $C(u)$ 在仿射变换下是不变的,即 $A[C(u)] = \sum_i A[P_i] R_{i,k}(u)$. NURBS 曲线在仿射变换下只改变控制点,权因子不变.

2° 透视变换　若投影中心为 C,透视投影平面由点 Q 和平面法矢量 N 定义,

则对一点 X 的透视投影变换是

$$\pi(X) = (1-\alpha)X + \alpha C,\quad \alpha = \frac{(X-Q)\cdot N}{(X-C)\cdot N}.$$

而对 NURBS 曲线的透视投影变换是

$$\pi[C(u)] = \frac{\sum_{i=0}^{n} w_i^* \pi[P_i] B_{i,k}(u)}{\sum_{i=0}^{n} w_i^* B_{i,k}(u)},$$

式中 $\pi[P_i]$ 表示对控制点作投影变换，$w_i^* = w_i(P_i - C)\cdot N$.

上式表明透视投影变换只改变控制点和权因子，并不改变 NURBS 基函数.

2. NURBS 曲线的导矢计算

把(2-22)式的分子分母分别记为

$$A(u) = \sum_{i=0}^{n} w_i P_i N_{i,k}(u),\quad w(u) = \sum_{i=0}^{n} w_i N_{i,k}(u),$$

于是(2-22)式可写为

$$C(u) = \frac{A(u)}{w(u)}.$$

对 $A(u) = w(u)C(u)$ 两边求导得

$$C'(u) = \frac{A'(u) - w'(u)C(u)}{w(u)},$$

从而有

$$C^{(r)}(u) = \frac{A^{(r)}(u) - \sum_{i=0}^{r}\binom{r}{i} w^{(i)} C^{(r-i)}(u)}{w(u)}.$$

特别地，曲线端点的切矢为

$$C'(0) = \frac{k}{u_{k+1}}\frac{w_1}{w_0}(P_1 - P_0),$$

$$C'(1) = \frac{k}{1-u_n}\frac{w_{n-1}}{w_n}(P_n - P_{n-1}).$$

3. 反求 NURBS 曲线的控制点

已知三次 NURBS 曲线的型值点 Q_i 及其相应的权因子 $w_i(i=0,1,\cdots,n)$，求相应 NURBS 曲线的控制点 P_j 及其权因子 $w_j^*(j=0,1,\cdots,n+2)$，下面讨论如何求 w_j^* 和 P_j.

(1) 求权因子 w_j^*　求一条 NURBS 曲线 $w(u)$，使得

$$w_i = w(u_{i+3}) = \sum_{j=0}^{n+2} N_{j,3}(u_{i+3}) w_j^*,\ i = 0,1,\cdots,n, \tag{2-23}$$

并使 w_j^* 非负. 为此构造以下二次规划问题：

$$
\begin{cases}
\min f(\overline{w}) = \overline{w}^{\mathrm{T}}\overline{w}, \\
w_i = \sum_{j=0}^{n+2} N_{j,3}(u_{i+3}) w_j^*, i = 0,1,\cdots,n, \\
w_j^* \geqslant 0, j = 0,1,\cdots,n+2.
\end{cases}
\tag{2-24}
$$

上式中 $\overline{w} = ((w_0^* - w_a),(w_1^* - w_a),\cdots,(w_{n+2}^* - w_a))^{\mathrm{T}}, w_a = \sum_{i=0}^{n} w_i/(n+1)$；在此假定 w_j^* 是三次 NURBS 曲线 $w(u)$ 上的点. 由于 $f(\overline{w})$ 矩阵是正定的，且有初值，因此可得到(2-23) 和(2-24) 式的最优解.

(2) 求控制点 P_j　由于三次 NURBS 曲线方程是

$$
C(u) = \sum_{j=0}^{n+2} w_j P_j N_{j,3}(u) \Big/ \sum_{j=0}^{n+2} w_j N_{j,3}(u),\quad 0 \leqslant u \leqslant 1,
$$

由前段的讨论已知权 w_j 的求法，现在只需将型值点代入上式，并增加两个边界条件，即可唯一地确定 P_j. 对于给定的型值点 Q_i，可以找到一条三次 NURBS 曲线，使得它在参数值为 u_{i+3} 时通过 Q_i，即有

$$
R_i = \sum_{j=0}^{n+2} \overline{X}_j N_{j,3}(u_{i+3}),\quad i = 0,1,\cdots,n,
\tag{2-25}
$$

式中 $\overline{X}_j = w_j^* P_j, R_i = Q_i \sum_{j=0}^{n+2} w_j N_{j,3}(u_{i+3}) = Q_i w_i$. (2-25) 式中只有 $(n+1)$ 个方程，但有 $(n+3)$ 个变量，需补充两个边界条件. 已知端点切矢量，$C'(u_0) = T_0$，$C'(u_{n+2k}) = T_n$；其端点二阶导数矢量 $C''(u_0) = C''(u_{n+2k}) = 0$；在始端，$C'(u_0) = T_0 = (4-1)w_1(P_1 - P_0)/w_0$ 即有 $w_1(P_1 - P_0) = T_0 w_0/3$；在末端，$C'(u_{n+2k}) = T_n = (4-1)w_{n+1}(P_{n+2} - P_{n+1})/w_{n+2}$，即有 $w_{n+1}(P_{n+2} - P_{n+1}) = T_n w_{n+2}/3$. 由始、末端切矢量建立的二个方程和(2-25) 式组合即可求得 $(n+3)$ 个控制点位置矢量 P_j.

反求控制点的过程中会有几种特殊情况，需要认真处理.

1° 权因子很小. 上述算法求出的权 w_j 均大于零，但可能很小，甚至为零. 此时可以将(2-24) 式目标函数改写为

$$
f(w) = \sum_{i=0}^{n+2} a_i (w_i - w_a)^2,
\tag{2-26}
$$

并取初值 $a = 1, i = 0,1,\cdots,n+2$；当对某个 j，有 $w_j \leqslant \epsilon$ 时，即令 $a'_j = kd$，并用 a'_j 代替(2-26) 式中的 a_i. 重新解此二次规划问题，通常取 $k = 10, \epsilon = 10^{-5}$（$\epsilon$ 为容差）.

2° 二重节点. 用向心模型计算节点矢量，当相邻两个型值点相等，即 $(P_i w_i) = P_{i+1} w_{i+1}$，会出现二重节点 $u_{i+k} = u_{i+k+l}$，这样，(2-25) 式中相应于 R_i 和 R_{i+1} 的两个方程完全相同，系数矩阵的秩减少 1，所以必须在方程组中补充一个方程才能使其有唯一解. 为此可补充一个在二重节点处的切矢量方程 $C'(t_{i+k}) = P'_i$，并用它取代原线性方程组中的 $R_i = \sum_{j=0}^{n+2} \overline{X} N_{j,3}(u_{i+3})$ 式. P_{ij}^* 可以事先给定，也可根据型值点，

利用数值微分法计算出来.

3° 三重节点. 当 $u_{i+k}=u_{i+k+1}=u_{i+k+2}$ 时即出现三重节点的情况. 此时(2-25)式的秩减少 2, 曲线在 u_{i+k} 处为 C 连续, 因此可补充两个表示该点处左导数和右导数的方程:

$$P'_i = C'(u_{ik}-0),\quad P'_{i+2}=C'(u_{i+k+2}+0),$$

并用它们取代(2-25)式中的 R_i 和 R_{i+2} 这两个方程, 使(2-25)满秩, 从而可得到唯一解.

4. 圆锥曲线的 NURBS 表示

若特征多边形的顶点为 P_i、P_{i+1}、P_{i+2}, 节点矢量为 $(u_i,u_{i+1},\cdots,u_{i+5})$, 且 $u_i=u_{i+1}=u_{i+2}<u_{i+3}=u_{i+4}=u_{i+5}$, 则用二次 NURBS 曲线 $C(u)=\sum_{i=0}^{n}P_iR_{i,3}(u)$ 表示圆锥曲线的充要条件是,

1° $\dfrac{w_i[(u_{i+4}-u_{i+3})w_{i+1}+(u_{i+3}-u_{i+2})w_{i+2}]}{w_{i+2}[(u_{i+3}-u_{i+2})w_i+(u_{i+2}-u_{i+1})w_{i+1}]}=\dfrac{|P_{i+2}-P_{i+1}|}{|P_{i+1}-P_i|}$,

2° $\dfrac{|[(u_{i+4}-u_{i+3})w_i-(u_{i+2}-u_{i+1})w_{i+2}]w_{i+1}P_{i+1}+aw_{i+2}P_{i+2}-bw_iP_i|^2}{|P_{i+1}-P_i|^2}$

$$=4bw_i^2w_{i+1}^2(u_{i+3}-u_{i+1})(u_{i+4}-u_{i+2})/a,$$

式中, $a=(u_{i+3}-u_{i+2})w_i+(u_{i+4}-u_{i+1})w_{i+1}$, $b=(u_{i+4}-u_{i+3})w_{i+1}+(u_{i+3}-u_{i+2})w_{i+2}$, $u_{i+2}<u_{i+3}$, 并且有心圆锥曲线的半径为

$$r_0=\frac{2(w_{i+1}w_{i+2})^2(u_{i+3}-u_{i+1})(u_{i+3}-u_{i+2})(u_{i+4}-u_{i+2})|P_{i+2}-P_{i+1}|^3}{b^3w_i|P_{i+1}-P_i|\times|P_{i+2}-P_{i+1}|},$$

圆锥曲线的形状因子

$$c_{sf}=\frac{(u_{i+3}-u_{i+1})(u_{i+4}-u_{i+2})w_{i+1}^2}{[(u_{i+3}-u_{i+2})w_i+(u_{i+2}-u_{i+1})w_{i+1}][(u_{i+4}-u_{i+3})w_{i+1}+(u_{i+3}-u_{i+2})w_{i+2}]}.$$

当 $c_{sf}<1$, $C(u)$ 为椭圆(圆是椭圆特例); $c_{sf}=1$, $C(u)$ 为抛物线; $c_{sf}>1$, $C(u)$ 为双曲线, 对应图形如图 2-15 所示.

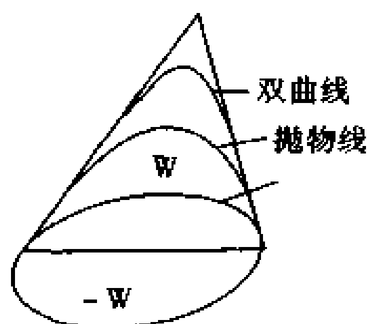


图 2-15

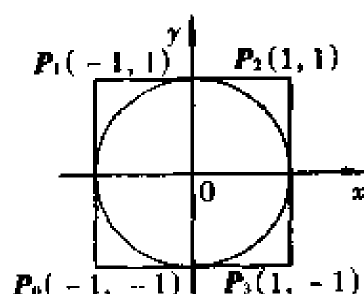


图 2-16

如图 2-16 所示, 若控制多边形为正方形, 节点矢量为 $(u_0,u_1,\cdots,u_5)$ 和权因子 w_0、w_1 和 w_2 满足:

$$\frac{(u_3-u_1)(u_4-u_2)}{(u_4-u_2)(u_2-u_1)}=2,$$

当$\frac{w_1}{w_2}=\frac{u_3-u_2}{u_4-u_3}, \frac{w_0}{w_1}=\frac{u_2-u_1}{u_4-u_3}$时,则二次周期性 NURBS 曲线

$$C_i(u)=\sum_{j=0}^{2} w_j P_{i+j} N_{j,3}(u) \Big/ \sum_{j=0}^{2} w_j N_{j,3}(u),$$

$$i=0,1,2,3; u \in [u_2,u_3]$$

构成一个精确的单位圆,其中 $u_0 \leqslant u_1 < u_2 < u_3 < u_4 \leqslant u_5$,且当 $P_5 \geqslant 4$时,$P_s=P_s-4$.

5. NURBS 曲线的修改

NURBS 曲线的修改有多种形式,常用的有修改权因子、修改控制点和分割 NURBS 曲线等.

(1) 权因子的修改及其几何意义

对于 k 次 NURBS 曲线 $C(u)=\sum_{i=0}^{n} P_i R_{i,k}(u)$,为了保证 $R_{i,k}(u)$ 非负,则要求权因子都大于零,从 $R_{i,k}(u)$ 可见权因子 w_i 的变化仅影响(u_i,u_{i+k+1})节点区间的曲线. 如图 2-17 中,$S=C(u;w_i=0)$;$M=C(u;w_i=1)$;$S_i=C(u;w_i \overline{\in} [0,1])$,通过简单置换得

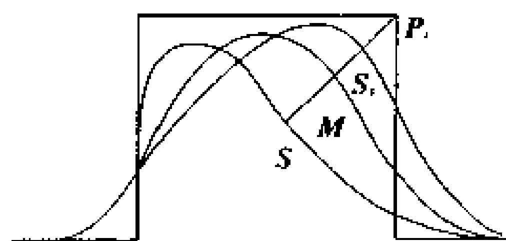


图 2-17

$$M=(1-t)S+tP_i, \quad S_i=(1-v)S+vP_i,$$

其中

$$t=\frac{N_{i,k}(u)}{\sum_{i \neq j=0}^{n} w_j N_{j,k}(u)+N_{i,k}(u)},$$

$$v=\frac{w_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)}=R_{i,k}(u).$$

沿着 SP_i 直线,P_i、S_i、M、S 四点的交叉比例是$\frac{(1-t)}{t}:\frac{(1-v)}{v}=w_i$.

1° 当w_i增加(减小)时,v 也增加(减少),曲线被拉向(拉开)P_i 点.

2° 当S_i移动,即产生一条过 P_i 点的直线时,曲线形状沿此直线变化,即曲线的变化是可预见的.

3° 当S_i靠近P_i点时,即 v 趋于 1, w_i 趋于无穷,此时要防止 w_i 上溢.实践证明,当 $w_i=100$ 时,曲线 S_i 就会很靠近 P_i,而不必取 w_i 太大.

修改过程是拾取曲线上一点 S_i,指定曲线变化方向(如以 P_i 和 S_i 的连线为变化方向),同时指定曲线的变化距离 d,则位置变为 S'_i,$d=|S'_i-S_i|$,现在要求 w'_i.

$$w'_i = \frac{1-t}{t} : \frac{1-v'}{v'} = \frac{P_iM}{SM} : \frac{P_iS'_i}{S_iS'_i} = \frac{P_iM}{SM} : \frac{P_iS_i \mp d}{SS_i \pm d}.$$

代入相应函数后 $w'_i = w_i\left[1 \pm \frac{d}{R_{i+k}(u)(P_iS_i \mp d)}\right]$，$w_i$ 是 P_i 点相应的权因子.

上式中，曲线拉向 P_i 点取"+"号，曲线拉开 P_i 点取"-"号. 当 u 趋于 u_i 或 u_{i+k+1} 时，w'_i 趋于无穷，$R_{i,k}(u)$ 趋于零，此处是某个控制点作为修改方向. 若不是以某个控制点，而是曲线上的某个点，则需要计算修改方向. 有了 w'_i 代入相应的公式生成新的曲线，即可满足用户要求.

(2) 修改控制点

对于曲线(2-22)，若给定曲线上一点 Q，或某个节点参数值 $\bar{u} \in [u_i, u_{i+k+1}]$、修改方向矢量 V 以及修改距离 d，要求计算新的控制顶点 P_i^*. 由

$$Q = P_0R_{0,k}(\bar{u}) + \cdots + (P_i + \alpha V)R_{i,k}(\bar{u}) + \cdots + P_nR_{n,k}(\bar{u}),$$

$$P_i^* = P_i + \alpha V, \; |P_i^* - P_i| = d = \alpha |V| R_{i,k}(u), \quad \alpha = \frac{d}{|V| R_{i,k}(\bar{u})}$$

得 α，即可求出新的控制点位置矢量，$P_i^* = P_i + \alpha V$. 这里 P_i 下标 i 的选取是由 $R_{i,k}(u)$ 取最大值点的参数与 $\bar{u}$ 距离最近而定.

更好的方法是同时考虑 $\bar{u}$ 处相邻的两个基函数对 α 值的影响，此时修改后的曲线形状较为对称. 当然在实际应用中也有拾取某个控制点 P_i 并把它改到 P_i^* 的直接情况(如用橡皮筋拖动 P_i 点到 P_i^*)，对此不必用上述方法计算.

(3) NURBS 曲线的分割

对此相当于在节点 u_i 和 u_{i+1} 之间插入一个节点 u^* 至满重复度，进而需求出权因子 w_j^* 和控制点 P_j^*. 从前几节的介绍可推出，

$$w_j^* = \alpha_j w_j + (1-\alpha_j)w_{j-1},$$

$$P_j^* = \frac{\alpha_j w_j P_j + (1-\alpha_j)w_{j-1}P_{j-1}}{\alpha_j w_j + (1-\alpha_j)w_{j-1}}.$$

其中 $\alpha_j = \begin{cases} 1, & j \leqslant i-k, \\ (u^* - u_j)/(u_{j+k} - u_j), & i-k+1 \leqslant j \leqslant i, \\ 0, & j \geqslant i+1. \end{cases}$

分割的终止条件一般取在 u^* 处有 $(k+1)$ 个重节点. 上述是执行一步插入节点的情形，多步情形的 α_j 见 B 样条曲线的 Oslo 算法.

2.4.2 非均匀有理 B 样条(NURBS) 曲面

1. NURBS 曲面的定义

由双参数变量分段有理多项式定义的 NURBS 曲面是

$$S(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{ij}P_{ij}N_{i,p}(u)N_{j,q}(v)}{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{ij}N_{i,p}(u)N_{j,q}(v)}, \tag{2-27}$$

式中 P_{ij} 是矩形域上特征网格控制点列，w_{ij} 是相应控制点的权因子，$N_{i,p}(u)$ 和 $N_{j,q}(v)$ 分别是 p 次和 q 次的B样条基函数，它们是在节点矢量 $U=(u_0,u_1,\cdots,u_{m+p+1})$ 和 $V=(v_0,v_1,\cdots,v_{n+q+1})$ 上定义的.若令

$$R_{ij}(u,v)=\frac{w_{ij}N_{i,p}(u)N_{j,q}(v)}{\sum_{x=0}^{m}\sum_{y=0}^{n}w_{xy}N_{x,p}(u)N_{y,q}(v)}$$

$R_{ij}(u,v)$ 是NURBS曲面的分段有理基函数.

若在非均匀参数轴上定义的节点矢量 U、V 具有下述形式，

$$U=(\underbrace{0,0,\cdots,0}_{p+1},u_{p+1},\cdots,u_{m+p-1},\underbrace{1,1,\cdots,1}_{p+1}),$$

$$V=(\underbrace{0,0,\cdots,0}_{q+1},,v_{q+1},\cdots,v_{n+q-1},\underbrace{1,1,\cdots,1}_{q+1}),$$

则由 U、V 定义的曲面的边界完全由顶点和权因子决定.通常设定权因子 w_{00}、w_{0n}、w_{m0}、$w_{nm}>0$，$w_{ij}\geqslant 0(i=1,2,\cdots,m-1;j=1,2,\cdots,n-1)$，这样可保证基函数为非负.从 $R_{ij}(u,v)$ 的定义可知，w_{ij} 仅影响区间 $[u_i,u_{i+p+1})$、$[v_j,v_{j+q+1})$ 范围内的曲面.节点矢量的定义对曲面的定义和修改也起到重要作用，定义方法可参见非均匀有理B样条曲线的介绍.

2. 插入控制点

若原来定义的曲面控制点网格是 $P_{ij}(i=0,1,\cdots,m;j=0,1,\cdots,n)$，现插入一个控制点 Q 及其权因子 w，定位于 P_{ij} 和 $P_{i+1,j+1}$ 之间，此时相当于在 u、v 两个方向上插入控制点，即有

$$Q_u=\frac{(1-\alpha)w_{ij}P_{ij}+\alpha w_{i+1,j}P_{i+1,j}}{(1-\alpha)w_{ij}+\alpha w_{i+1,j}},$$

$$\alpha=\frac{w_{ij}\mid Q_u-P_{ij}\mid}{w_{ij}\mid Q_u-P_{ij}\mid+w_{i+1,j}\mid P_{i+1,j}-Q_u\mid}.$$

在 u 向插入的节点应是： $u=u_{i+1}+\alpha(u_{i+p+1}-u_{i+1})$.

类似地可得到 v 向插入控制点的公式

$$Q_v=\frac{(1-\beta)w_{ij}P_{ij}+\beta w_{i,j+1}P_{i,j+1}}{(1-\beta)w_{ij}+\beta w_{i,j+1}},$$

$$\beta=\frac{w_{ij}\mid Q_v-P_{ij}\mid}{w_{ij}\mid Q_v-P_{ij}\mid+w_{i,j+1}\mid P_{i,j+1}-Q_v\mid},$$

在 v 向插入的节点应是： $v=v_{j+1}+\beta(v_{j+q+1}-v_{j+1})$.

若没有给出 Q 点相应的权因子 w，可以通过相邻的 w_{ij} 和 $w_{i+1,j+1}$ 线性插值得到.

3. 修改权因子

权因子 w_{ij} 的修改影响 $[u_i,u_{i+p+1})\times[v_j,v_{j+q+1})$ 矩形区域上的曲面，$u\in[u_i,u_{i+p+1})$，$v\in[v_j,v_{j+q+1})$，改变 w_{ij} 的几何意义和修改曲线 w_i 的几何意义相同，其中记 $S=S(u,v;w_{ij}=0)$；$M=S(u,v;w_{ij}=1)$；$S_{ij}=S(u,v;w_{ij}\neq 0,1)$.$M$ 和 S_{ij} 可表示为

$$M=(1-a)S+aP_{ij};\quad S_{ij}=(1-b)S+bP_{ij},$$

其中

$$a = \frac{N_{i,p}(u)N_{j,q}(v)}{\sum_{i\neq x=0}^{m}\sum_{\neq y=0}^{n} w_{xy}N_{x,p}(u)N_{y,q}(v) + N_{i,p}(u)N_{j,q}(v)},$$

$$b = \frac{w_{ij}N_{i,p}(u)N_{j,q}(v)}{\sum_{x=0}^{m}\sum_{y=0}^{n} w_{xy}N_{x,p}(u)N_{y,q}(v)} = R_{ij}(u,v).$$

由上式可知$\frac{(1-a)}{a}:\frac{(1-b)}{b} = w_{ij}$,这实际上是四点 $\boldsymbol{P}_{ij}$、S、$\boldsymbol{M}$、$\boldsymbol{S}_{ij}$ 的交叉比,并可知:

(1) 当 w_{ij} 增加(减少),曲面被拉向(拉开)$\boldsymbol{P}_{ij}$ 点.

(2) $\boldsymbol{S}_{ij}$ 仅沿直线 $\boldsymbol{PP}_{ij}$ 移动,若要使 $\boldsymbol{S}_{ij}$ 趋向 $\boldsymbol{P}_{ij}$,则要求 w_{ij} 趋向无穷.

(3) 若要求在 $\boldsymbol{P}_{ij}$ 点把曲面沿 SP_{ij} 直线拉向(拉开)指定距离 d,则要重新计算权因子 w_{ij},这和推导 NURBS 曲线的 w_i 相似,权因子应修改为

$$w'_{ij} = w_{ij}\left[1 \pm \frac{d}{R_{ij}(u,v)(P_{ij}S_{ij} - d)}\right],$$

其中 $\boldsymbol{S}_{ij}$ 是曲面上的一点,要把曲面拉向 $\boldsymbol{P}_{ij}$ 取"+"号,拉开 $\boldsymbol{P}_{ij}$ 取"-"号.

4. 修改控制点

与曲线情形类似,给定曲面上点 $\boldsymbol{Q}$,要计算的相应参数值是$(\bar{u},\bar{v}) \in [u_{i,i+p+1}) \times [v_{j,j+q+1})$,曲面变化的方向矢量 $\boldsymbol{T}$ 及其变化距离d,要求计算相应控制顶点 $\boldsymbol{P}_{ij}$ 的新位置$\boldsymbol{P}_{ij}^*$.因

$$\boldsymbol{Q} = \boldsymbol{P}_{00}R_{00}(\bar{u},\bar{v}) + \cdots + (\boldsymbol{P}_{ij} + \alpha\boldsymbol{T})R_{ij}(\bar{u},\bar{v}) + \cdots + \boldsymbol{P}_{mn}R_{mn}(\bar{u},\bar{v}),$$

令 $\alpha = \frac{d}{|\boldsymbol{T}|R_{ij}(\bar{u},\bar{v})}$,则

$$\boldsymbol{P}_{ij}^* = \boldsymbol{P}_{ij} + \alpha\boldsymbol{T}.$$

$\boldsymbol{P}_{ij}$ 的下标的选取是由 $R_{ij}(u,v)$ 的最值所对应参数$(\hat{u},\hat{v})$与$(\bar{u},\bar{v})$最近的 $R_{ij}(u,v)$ 而定.

2.5　孔斯曲面

通常意义下的曲线、曲面插值,都是插值有限个离散点,而孔斯(Coons)曲面是插值围成封闭区域的四条边界曲线.这种插值方法,首先由孔斯提出,称为**超限插值法**.它在有限元分析、偏微分方程数值解等领域有重要应用.

2.5.1　线性孔斯曲面

线性孔斯曲面,也称之为简单曲面,是通过四条边界曲线构成的曲面.设二元函数 $\boldsymbol{p}(u,w) \in C[0,1;0,1]$,记 $\boldsymbol{p}(u,0)$、$\boldsymbol{p}(u,1)$、$\boldsymbol{p}(0,w)$、$\boldsymbol{p}(1,w)$ 为矩形域$[0,1]\times[0,1]$上给定的四条边界曲线,简记为 $\boldsymbol{p}_{u0}$,$\boldsymbol{p}_{u1}$,$\boldsymbol{p}_{0w}$,$\boldsymbol{p}_{1w}$,另外记$\frac{\partial p(u,w)}{\partial u} = \boldsymbol{p}_{uw}^u$

等.

在 u 向进行线性插值,得到直纹面为

$$S_1(u,w) = (1-u)\boldsymbol{p}_{0w} + u\boldsymbol{p}_{1w};$$

在 w 向进行线性插值,得到直纹面为

$$S_2(u,w) = (1-w)\boldsymbol{p}_{u0} + u\boldsymbol{p}_{u1};$$

如图 2-18 所示.

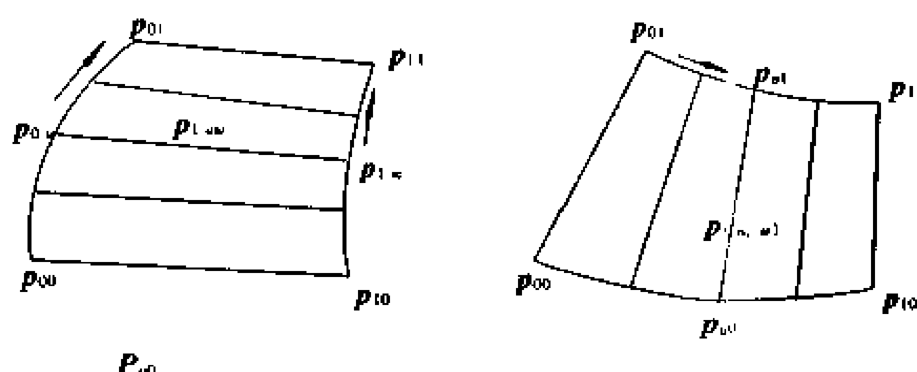


图 2-18

若把这两张直纹面叠加可得到一张新曲面 $S_3(u,w)$,使其的边界正好就是所不需要的线性插值边界,为此可得到在 w 方向上进行线性插值构造 $\boldsymbol{S}_3(u,w)$ 的公式:

$$\begin{aligned} S_3(u,w) &= (1-w)[(1-u)\boldsymbol{p}_{00} + u\boldsymbol{p}_{10}] + w[(1-u)\boldsymbol{p}_{01} + u\boldsymbol{p}_{11}] \\ &= (1-u)(1-w)\boldsymbol{p}_{00} + u(1-w)\boldsymbol{p}_{10} + (1-u)w\boldsymbol{p}_{01} + uw\boldsymbol{p}_{11}, \end{aligned}$$

则用四条边界曲线构造的曲面 $\boldsymbol{S}(u,w) = \boldsymbol{S}_1(u,w) + \boldsymbol{S}_2(u,w) - \boldsymbol{S}_3(u,w)$ 可写成

$$\boldsymbol{S}(u,w) = [1-u \quad u]\begin{bmatrix} \boldsymbol{p}_{0w} \\ \boldsymbol{p}_{1w} \end{bmatrix} + [\boldsymbol{p}_{u0} \quad \boldsymbol{p}_{u1}]\begin{bmatrix} 1-w \\ w \end{bmatrix} - [1-u \quad u]\begin{bmatrix} \boldsymbol{p}_{00} & \boldsymbol{p}_{01} \\ \boldsymbol{p}_{10} & \boldsymbol{p}_{11} \end{bmatrix}\begin{bmatrix} 1-w \\ w \end{bmatrix}. \tag{2-28}$$

对于该曲面,当 $u=0, u=1, w=0, w=1$ 时,对应的四条边界曲线即为已知 $\boldsymbol{p}_{u0}$, $\boldsymbol{p}_{u1}, \boldsymbol{p}_{0w}, \boldsymbol{p}_{1w}$.(2-28) 式所用的调配函数是 u、$(1-u)$、w、$(1-w)$,即为用均匀线性插值法构造曲面.(2-28) 式改写成矩阵式:

$$\boldsymbol{S}(u,v) = -[-1 \quad u \quad 1-u]\begin{bmatrix} 0 & \boldsymbol{p}_{u0} & \boldsymbol{p}_{u1} \\ \boldsymbol{p}_{0w} & \boldsymbol{p}_{00} & \boldsymbol{p}_{01} \\ \boldsymbol{p}_{1w} & \boldsymbol{p}_{10} & \boldsymbol{p}_{11} \end{bmatrix}\begin{bmatrix} -1 \\ w \\ 1-w \end{bmatrix}, \quad u,w \in [0,1]. \tag{2-29}$$

2.5.2　第二类孔斯曲面

第二类孔斯曲面不仅插值于曲面的四条边界,而且插值于给定边界的斜率,这种曲面和第一类孔斯曲面一样,也可看作是三张面的组合.即 $\boldsymbol{S}_1(u,w) + \boldsymbol{S}_2(u,w)$

$-S_3(u,w)$. 若已知该曲面的四条边界 $p_{u0}, p_{u1}, p_{0w}, p_{1w}$, 边界上的导数矢量分别是 p_{u0}^w、p_{u1}^w、p_{0w}^u、p_{1w}^u. 选用三次埃尔米特插值基函数, 以 p_{0w}、p_{1w} 为基线, p_{0w}^u、p_{1w}^u 为边界切矢, 得曲面 $S_1(u,w)$, 若以 p_{u0}、p_{u1} 为基线, p_{u0}^w、p_{u1}^w 为边界切矢, 得曲面 $S_2(u,w)$, 当 $S_1(u,w)$ 和 $S_2(u,w)$ 叠加后多余了边界信息和边界切矢, 由多余信息构造张量积曲面 $S_3(u,w)$, 则满足已知边界和边界切矢的曲面为

$$S(u,w) = -[-1\ F_0(u)\ F_1(u)\ G_0(u)\ G_1(u)]\begin{bmatrix} 0 & p_{u0} & p_{u1} & p_{u0}^w & p_{u1}^w \\ p_{0w} & p_{00} & p_{01} & p_{00}^w & p_{01}^w \\ p_{1w} & p_{10} & p_{11} & p_{10}^w & p_{11}^w \\ p_{0w}^u & p_{00}^u & p_{01}^u & p_{00}^{uw} & p_{01}^{uw} \\ p_{1w}^u & p_{10}^u & p_{11}^u & p_{10}^{uw} & p_{11}^{uw} \end{bmatrix} \cdot \begin{bmatrix} -1 \\ F_0(w) \\ F_1(w) \\ G_0(w) \\ G_1(w) \end{bmatrix}, u,v \in [0,1]. \tag{2-30}$$

记

$$B = \begin{bmatrix} p_{00} & p_{01} & p_{00}^w & p_{01}^w \\ p_{10} & p_{11} & p_{10}^w & p_{11}^w \\ p_{00}^u & p_{01}^u & p_{00}^{uw} & p_{01}^{uw} \\ p_{10}^u & p_{11}^u & p_{10}^{uw} & p_{11}^{uw} \end{bmatrix} \tag{2-31}$$

B 称为**信息矩阵**. 它可以分为四个子块

$$\begin{bmatrix} \text{角点型值} & \text{解点处 } w \text{ 向切向值} \\ \text{角点处 } u \text{ 向切向值} & \text{角点处的扭矢值} \end{bmatrix}.$$

通常所指的孔斯曲面就是指上述双三次孔斯曲面. 上述 G_i、F_i 是三次埃尔米特调配函数($i=0,1$), 其中 p^{uw} 混合切矢度量了曲面 $S(u,w)$ 在 u 向的切矢沿 w 向的变化, 也是曲面扭曲程度的一种度量. 若 p^{uw} 是 u 和 w 的连续函数, 则有 $p^{uw}=p^{wu}$, p^{uw} 也常称之为扭矢. 经直接验证可知 $S(u,w)$ 在单位正方形的四条边上插值 $p(u,w)$ 及其法向导数的充要条件是, 协调条件(或称相容条件):

$$\frac{\partial^2 p}{\partial u \partial w} = \frac{\partial^2 p}{\partial w \partial u}$$

在单位方域的四个顶点成立.

在实际问题中, 常常仅知道 p 及 p 的一阶导数值, 至于 p 的其它性质(如扭矢)较难掌握. 因此, p 在四个角顶的协调性是必须验证的, 鉴于此, 人们引进了各种修正的孔斯插值曲面, 它们无需检查协调条件.

下面介绍几种常用的修正的插值曲面.

1. 格利哥里曲面

在信息矩阵(2-31)中, 用如下的变形扭转子块

$$
D = \begin{bmatrix} \dfrac{u p_{00}^{wu} + w p_{00}^{uw}}{u + w} & \dfrac{-u p_{01}^{wu} + (w-1) p_{01}^{uw}}{-u + w - 1} \\ \dfrac{(1-u) p_{10}^{wu} + w p_{10}^{uw}}{1 - u + w} & \dfrac{(u-1) p_{11}^{wu} + (w-1) p_{11}^{uw}}{u + w - 2} \end{bmatrix} \tag{2-32}
$$

替代(2-31)中的扭转子块，可得格利哥里(Gregory)曲面：

$$
\begin{aligned}
\hat{S}(u,w) = S(u,w) &- \frac{u^2(u-1)^2 w(w-1)^2}{u+w}[p_{00}^{wu} - p_{00}^{uw}] - \\
&\frac{u^2(u-1)^2 w(w-1)^2}{-w+u+1}[p_{01}^{wu} - p_{01}^{uw}] - \\
&\frac{u^2(u-1)^2 w^2(w-1)}{-w+u-1}[p_{10}^{wu} - p_{10}^{uw}] - \\
&\frac{u^2(u-1)^2 w^2(w-1)}{u+w-2}[p_{11}^{wu} - p_{11}^{uw}],
\end{aligned} \tag{2-33}
$$

其中 $S(u,w)$ 为(2-31)式.

格利哥里曲面是属于 $C^{1,1}$ 的光滑曲面，若 $p(u,w)$ 在标准正方形的四个顶点处满足协调条件，则格利哥里曲面就是孔斯曲面.

2. 布朗(Brown)曲面

注意到第二类孔斯曲面的不相容性起因于张量积曲面 $S_3(u,w)$，布朗直接用 $S_1(u,w)$ 和 $S_2(u,w)$ 的凸组合构造孔斯曲面而不用 S_3. 布朗曲面由下式表示：

$$
S(u,w) = \alpha(u,w) S_1(u,w) + \beta(u,w) S_2(u,w),
$$

其中

$$
\alpha(u,w) = \frac{w^2(1-w)^2}{u^2(1-u)^2 + w^2(1-w)^2}, \quad \beta(u,w) = \frac{u^2(1-u)^2}{u^2(1-u)^2 + w^2(1-w)^2}
$$

$\alpha(u,w)$、$\beta(u,w)$ 起着调配函数的作用.

$S(u,w)$ 在正方形域的边界上插值 p 和它的法向导数值，不难验证 $S(u,w)$ 为双三次多次式.

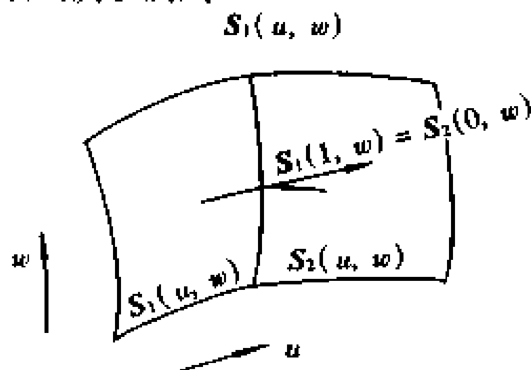


图 2-19

2.5.3　孔斯曲面片的拼接

如图 2-19 所示，两张孔斯曲面——$S_1(u,w)$ 和 $S_2(u,w)$ 拼接，要求在公共边界处达到 C^0、C^1 连续的条件是：C^0 连续要求两面片公共边界重叠，即有 $S_2(0,w) = S_1(1,w)$；达到 C^1 连续则要求 $S_2(0,w)$ 的切平面和 $S_1(1,w)$ 的切平面共面，且其法矢的方向保持一致. 设 $\mu(w)$ 为正的标量函数，则有 $S_2^u(0,w) \times S_2^w(0,w) = \mu(w) S_1^u(1,w) \times S_1^w(1,w)$，$u$、$w \in [0,1]$；因 C^0 连续，有 $S_2^w(0,w) = S_1^w(1,w)$. 若设 $r(w)$ 为任意标量函数，则有

$$
S_2^u(0,w) = \mu(w) S_1^u(1,w) + r(w) S_1^w(1,w).
$$

此式的几何意义是，曲面片 $S_2(u,w)$ 在 u 向的切矢 $S_2^u(0,w)$ 位于曲面片 $S_1(u,w)$

在同一边界处的平面上. 对于特殊情况 $r(w)=0$,则 C^1 连续的条件为:$S_2^u(0,w)=\mu(w)S_1^\varphi(1,w)$. 由上述可知两张孔斯曲面片在公共边界 $S_2(0,w)=S_1(1,w)$ 上实现 C^1 连续的充要条件是 $S_2^u(0,w)\cdot(S_1^u(1,w)\times S_1^\varphi(1,w))=0$.

2.5.4 戈登曲面

戈登(Gordon)曲面是孔斯曲面的推广,是双向曲线网格的插值.

欲构造一张实用的曲面仅仅用四条边界曲线往往是不够的. 给定由两组曲线交织成的曲线网格,如图 2-20. 构造一张曲面 $g(u,v)$ 插值所有这些曲线. 并且给定曲线为曲面的两族等参数线 $g(u_i,v)$, $i=0,1,\cdots,m$ 和 $g(u,v_j)$, $j=0,1,\cdots,n$. 构造戈登曲面的思想与构造孔斯曲面片相同.

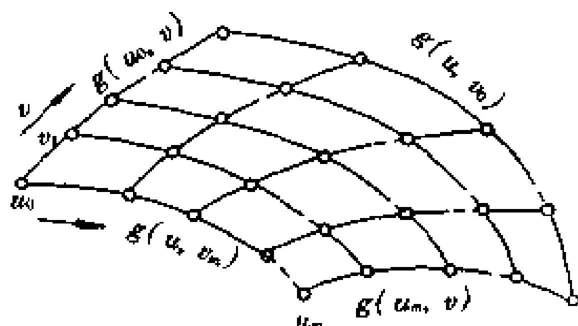


图 2-20

分别在 u 与 v 的两个分割 $\Delta_u: u_0<u_1<\cdots<u_m$, $\Delta_v: v_0<v_1<\cdots<v_n$ 上选取用于插值的两组单变量基函数:

$$\varphi_i(u),\quad i=0,1,\cdots,m \qquad 与 \qquad \varphi_j(v),\quad j=0,1,\cdots,n,$$

使满足于

$$\varphi_i(u_j)=\varphi_i(v_j)=\delta_{ij}=\begin{cases}1, & j=i,\\0, & j\neq i.\end{cases}$$

构造插值一族 v 线的曲面

$$q(u,v)=\sum_{i=0}^{m}g(u_i,v)\varphi_i(u),$$

又构造插值一族 u 线的曲面

$$r(u,v)=\sum_{j=0}^{n}g(u,v_j)\varphi_j(v),$$

再构造插值网格点阵的一张量积曲面

$$S(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}g(u_i,v_j)\varphi_i(u)\varphi_j(v).$$

戈登曲面为

$$\begin{aligned}g(u,v)&=q(u,v)+r(u,v)-s(u,v)\\&=\sum_{i=0}^{m}g(u_i,v)\varphi_i(u)+\sum_{j=0}^{n}g(u,v_j)\varphi_j(v)-\end{aligned}$$

$$\sum_{i=0}^{m}\sum_{j=0}^{n} g(u_i,v_j)\varphi_i(u)\varphi_j(v),$$

或可改写为

$$g(u,v) = -[-1\ \varphi_0(u)\ \cdots\ \varphi_m(u)]$$

$$\begin{bmatrix} 0 & g(u,v_0) & \cdots & g(u,v_n) \\ g(u_0,v) & g(u_0,v_0) & \cdots & g(u_0,v_n) \\ \vdots & \vdots & & \vdots \\ g(u_m,v) & g(u_m,v_0) & \cdots & g(u_m,v_n) \end{bmatrix} \begin{bmatrix} -1 \\ \varphi_0(v) \\ \vdots \\ \varphi_n(v) \end{bmatrix}.$$

在实际应用中，当 m、n 较小时，可选用伯恩斯坦基或拉格朗日基等. m、n 较大时，可选样条基. 特殊地，当 $m = n = 1$ 时，

$u_0 = v_0 = 0, u_1 = v_1 = 1, \varphi_0(u) = 1 - u,\ \varphi_1(u) = u,\ \varphi_0(v) = 1 - v,\ \varphi_1(v) = v,$
戈登曲面就成为双线性混合孔斯曲面片.

2.6　等距曲线和曲面

2.6.1　等距曲线

在曲面造型、数控加工中的刀路计算和机器人行进路径规划等问题中都需求等距曲线.

对于一般的平面正则曲线 $\boldsymbol{r}(t)$ 的**等距曲线**定义如下：

定义 2　$\boldsymbol{r}(t)$ 的等距线(offset)$\boldsymbol{r}_0(t)$ 定义为

$$\boldsymbol{r}_0(t) = \boldsymbol{r}(t) + d\boldsymbol{n}(t), t \in [0,1],$$

式中 $\boldsymbol{n}(t)$ 是 $\boldsymbol{r}(t)$ 上任一点的单位法线向量，d 是偏移有向距离，利用 d 的正负性可以确定等距线的正向或负向偏移.

定义 3　$\boldsymbol{r}_0(t)$ 在 t_c 处有尖点，若原曲线 $\boldsymbol{r}(t)$ 的曲率 $K(t_c) = -1/d$ 并且 $K'(t_c) \neq 0$. 当对任意 $t \in (0,1)$, $K(t) \neq -1/d$ 时，则等距线 $\boldsymbol{r}_0(t)$ 称为是非退化的.

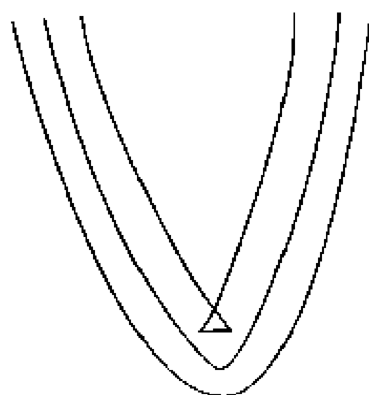


图 2-21

定义 4　等距线 $\boldsymbol{r}_0(t)$ 在 t_c 点处有异常点(extraordinary point)，若该点处 $K(t_c) = -1/d$ 且 $k'(t_c) = 0$, $k''(t_c) \neq 0$. 如图 2-21. 上述 k 是 $\boldsymbol{r}(t)$ 的曲率.

$\boldsymbol{r}(t)$ 与 $\boldsymbol{r}_0(t)$ 的几何量有下列关系：

设 $\boldsymbol{t}$、$\boldsymbol{n}$、k 分别是 $\boldsymbol{r}(t)$ 的单位切矢、单位法矢和曲率，则

$$\boldsymbol{t}_0 = \frac{1 + kd}{|1 + kd|}\boldsymbol{t},\quad \boldsymbol{n}_0 = \frac{1 + kd}{|1 + kd|}\boldsymbol{n},$$

$$k_0 = \frac{k}{|1 + kd|}$$

是 $\boldsymbol{r}_0(t)$ 上对应点处的单位切矢、单位法矢和曲率.

定义 5　距离函数 $\delta(p, C)$ 定义为一点 $\boldsymbol{p}$ 和一正

则参数曲线 $C = \{r(t) \mid t \in [0,1]\}$ 之间的距离.

$$\delta(p,C) = \inf_{t\in[0,1]} |p - r(t)|.$$

定理 3　等距线上点 $r_0(\tau)$ 和原曲线 $C = \{r(t) \mid t \in [0,1]\}$ 之间的距离 $\delta(r_0(\tau),C)$ 满足如下条件之一：

$$\delta(r_0(\tau),C) = |d|,$$

$$\delta(r_0(\tau),C) < |d|,$$

$\tau \in (t_k,t_{k+1})$，$\{t_k\}$，$k = 0,1,\cdots,N$ 是 $r_0(t)$ 上出现自交点的有序参数列. 其中 $t_0 = 0$，$t_n = 1$.

由等距线的定义知，有理曲线的等距线不一定是有理的，因此常用逼近方法表示等距线，下面是一种 NURBS 曲线的等距线求法，仍用 NURBS 曲线表示.

设 k 次 NURBS 曲线定义为

$$C(u) = \sum_{i=0}^{n} w_i P_i R_{i,k}(u),$$

其中 $R_{i,k}(u)$ 是 k 次有理 B 样条基函数，它由节点矢量 $U = (u_0,u_1,\cdots,u_{n+k+1})$ 决定. 记 $C(u)$ 的有向偏移距离为 d 的等距曲线为 $C_0(u)$，仍用 NURBS 表示，并且 $C(u)$ 与 $C_0(u)$ 具有相同的节点矢量和权因子. 求 $C_0(u)$ 表示式的一种方法是直接对 $C(u)$ 的控制顶点 $\{P_i\}$ 作偏移：$P_i^0 = P_i + \lambda_i n_i$，$\lambda_i$ 是待定系数，n_i 是 $C(u)$ 上某一点的单位法向，从而问题转化为确定 λ_i 和 n_i.

(1) 结点和控制结点　结点定义为：

$$\zeta_i = \frac{1}{k}\sum_{j=i+1}^{k} u_j, \quad j = 0,1,\cdots,n.$$

结点 $\{\zeta_i\}$ 反映了节点 $\{u_i\}_{i=0}^{n+k+1}$ 的分布状况. ζ_i 是距 $R_{i,k}(u)$ 最大值所对应的参数 $\hat{u}$ 最近的参数值. 控制结点定义为

$$\eta_i = \{u \mid |C(u) - P_i| = \min, u \in [u_k,u_{n+1}]\},$$

控制结点与控制顶点相对应，可用下述迭代过程求得，

$$\eta_i^{m+1} = \eta_i^m + \frac{C'(\eta_i^m)\cdot(P_i - C(\eta_i^m))}{|C'(\eta_i^m)|\cdot L},$$

式中 L 为 $C(u)$ 的近似弧长，迭代初值可取 ζ_i，即 $\eta_i^0 = \zeta_i$.

(2) 控制顶点 P_i^0，

$$P_i^0 = P_i + \lambda_i n(\eta_i),$$

$$\lambda_i = d(1 + K|P_i - C(\eta_i)|)$$

式中 $n(\eta_i)$ 表示 $C(u)$ 上参数值为 η_i 处的曲线的单位法矢，K 是曲线上 η_i 处的曲率，显然当 $C(u)$ 为直线或圆弧时，这样取 λ_i 可精确求出等距线.

(3) 误差分析　在非直线圆弧情形，$C_0(u)$ 只能近似表示 $C(u)$，逼近程度的误差界定有多种形式，一种方式是取 N 个样本点检查. 在 $[u_k,u_{n+1}]$ 中取 N 个点，令

$$t_i = i(u_{n+1} - u_k)/(N+1), \quad i = 1,2,\cdots,N,$$

给定迭代初值 $u_i^0 = u_i$，即可解出 $\bar{u}_i$，使 $C(\bar{u}_i)$ 为曲线 $C(u)$ 刻点 $C_0(u_i^0)$ 的局部最

近点.对每个 u_i 求出 $d_i = |C(u_i) + d\boldsymbol{n}(u_i) - C_0(\overline{u_i})|$,若所有的 d_i 都小于预先给定的精度 ϵ,则认为 $C_0(u)$ 满足需求.反之,对多个满足 $d_i > \epsilon$ 的 i,找出 u_i、u_{i+1},使得 $u_j \leqslant t_i < u_{j+1}$,并将 $u = (u_j + u_{j+1})/2$ 作为新节点插入到节点矢量 $\boldsymbol{U}$ 中,并求出 $C(u)$ 的表达式.如有多个 t_i 满足 $u_j \leqslant t_i < u_{j+1}$,则 u 只插入一次,并重新计算 $C_0(u)$.

2.6.2　等距面

1. 定义

与曲线等距线相比,自由曲面的等距面计算较为复杂.计算等距面的几何量也较为复杂.

定义 6　给定正则的参数曲面 $S = S(u,v)$,S 的等距曲面定义为

$$S_0(u,v) = S(u,v) + d\boldsymbol{n}(u,v),$$

式中 $\boldsymbol{n}(u,v) = \dfrac{\boldsymbol{r}_u \times \boldsymbol{r}_v}{|\boldsymbol{r}_u \times \boldsymbol{r}_v|}$ 为曲面 S 上一点 (u,v) 的单位法矢,d 为等距面的偏移量,d 的正负取值取决于偏移方向是指向曲面的外侧或内侧.

由于等距面表示的复杂性,许多情况下,只要把曲面的内在几何量计算出来,则可以不必写出等距面的表达式,也可作有关等距面的运算,如在曲面求交等操作中.对于正则曲面 $\boldsymbol{r} = \boldsymbol{r}(u,v)$,设定下列记号:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{r}_u \cdot \boldsymbol{r}_u & \boldsymbol{r}_u \cdot \boldsymbol{r}_v \\ \boldsymbol{r}_u \cdot \boldsymbol{r}_v & \boldsymbol{r}_v \cdot \boldsymbol{r}_v \end{bmatrix}$$

为曲面的第一基本矩阵,于是 $|\boldsymbol{G}| = |\boldsymbol{r}_u \times \boldsymbol{r}_v|^2$,记

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{n} \cdot \boldsymbol{r}_{uu} & \boldsymbol{n} \cdot \boldsymbol{r}_{uv} \\ \boldsymbol{n} \cdot \boldsymbol{r}_{uv} & \boldsymbol{n} \cdot \boldsymbol{r}_{vv} \end{bmatrix}$$

为曲面的第二基本矩阵.曲面上的高斯曲率 $\widetilde{K}$ 和中曲率定义为

$$\widetilde{K} = \frac{|\boldsymbol{D}|}{|\boldsymbol{G}|} = \frac{(\boldsymbol{n} \cdot \boldsymbol{r}_{uu})(\boldsymbol{n} \cdot \boldsymbol{r}_{vv}) - (\boldsymbol{n} \cdot \boldsymbol{r}_{uv})^2}{|\boldsymbol{r}_u \times \boldsymbol{r}_v|^2},$$

$$H = \frac{-2(\boldsymbol{n} \cdot \boldsymbol{r}_{uv})(\boldsymbol{r}_u \cdot \boldsymbol{r}_v) + (\boldsymbol{n} \cdot \boldsymbol{r}_{uu})(\boldsymbol{r}_v \cdot \boldsymbol{r}_v) + (\boldsymbol{n} \cdot \boldsymbol{r}_{vv})(\boldsymbol{r}_u \cdot \boldsymbol{r}_u)}{2|\boldsymbol{r}_u \times \boldsymbol{r}_u|^2},$$

从而等距面相应的几何量为

$$K_0 = \frac{K}{Kd^2 - 2Hd + 1},$$

$$H_0 = \frac{H + Kd}{Kd^2 - 2Hd + 1}.$$

与曲线情形类似,可利用曲面任一点的最小曲率来界定曲面上的奇异性.

与曲线情形一样,在几何造型中常常需要用与原曲面同类型的曲面表示等距面,由定义知,除平面、球面、圆环面、圆柱面等几种曲面外,有理曲面的等距面一般不再是有理的.因此,工程上常用逼近方法表示等距面.下面的算法用于计算和构造等距面.

2. 计算等距面的过程

设参数曲面的定义域是$[u_0,u_n]\times[v_0,v_m]$,初始化为Δu_0、ΔV_0及控制精度δ.

(1) 固定V_j,计算曲面上的点坐标$S(u_i,v_j)$及其法矢$n(u_i,v_j)$以及相应等距面上的点$S_0(u_i,v_j)$.

(2) 计算等距面上的新点$S_0(u_i+\Delta u,v_j)$.

(3) 分析用弦$S_0(u_i,v_j)S_0(u_i+\Delta u,v_j)$逼近相应曲线的误差$\varepsilon$,若$\varepsilon<\delta$,则$\Delta u=1.382\times\Delta u$;否则,$\Delta u=\Delta u\times 0.618$;$u_{i+1}=u_i+\Delta u$;若$u_i<u_n$转(2),否则转(4).

(4) 若$j>m$,则结束;否则,在u_i的可取值范围内均匀采样$u_0,u_1,\cdots,u_4$五点,且$u_4=u_n$,计算用弦$S(u_k,v_j)S(u_k,v_j+\Delta v)$逼近相应曲线的误差$\varepsilon$,$k=0,1,\cdots,4$;若$\varepsilon>\delta$,$\Delta v=0.618\times\Delta v$.重复本步工作;若$\varepsilon<\delta$,则做(5).

(5) 对上述五个采样点排序找出其中误差最大者ε_{max}及其左右两点$\varepsilon_{max}-1$和$\varepsilon_{max}+1$.$\varepsilon_{max}-1$、ε_{max}和$\varepsilon_{max}+1$三点将形成单峰区域,用数值规划法不难求出极大值处的u_i^*,并用u_i^*对应的Δv作为v向的步长,并取$v_{j+1}=v_j+\Delta v$,转(1).

等距面的离散点列生成后,可以用等参折线集或三角面片的形式输出该面.

2.7　三角域上的曲面表示

2.7.1　面积坐标和重心坐标

三角形abc所在平面上的一点p(如图2-22)可表示成

$$p=ua+vb+wc$$

其中$u+v+w=1$.u、v、w就是p点关于三角形abc的**重心坐标**.

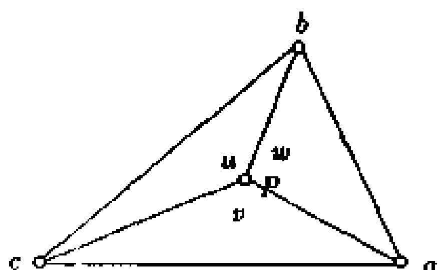


图2-22

当p点在平行于bc边的直线上移动时,它的u坐标不变.类似地,当P点分别沿ca边和$\overline{ab}$边移动时,分别有v坐标与w坐标不变.

重心坐标与面积坐标是一致的,即有

$$u=\frac{S_{\triangle pbc}}{S_{\triangle abc}},\quad v=\frac{S_{\triangle pca}}{S_{\triangle abc}},\quad w=\frac{S_{\triangle pab}}{S_{\triangle abc}},$$

$S_{\triangle abc}$表示$\triangle abc$的有向面积,按顶点字母顺序,逆时针旋转为正,顺时针旋转为负.其它,如$S_{\triangle pbc}$,的记号也如此.

2.7.2　三角域上的伯恩斯坦基函数

定义在三角域上的双变量的n次伯恩斯坦基函数由$(u+v+w)^n$的展开式各项组成.

$$(u+v+w)^n=\sum_{i+j+k=n}B_{i,j,k}^n(u,v,w),$$

伯恩斯坦基函数

$$B^n_{i,j,k}(u,v,w) = \frac{n!}{i!j!k!}u^i v^j w^k, \qquad 0 \leqslant u,v,w \leqslant 1,$$

其中，$i+j+k=n$，且 $i、j、k \geqslant 0$. 可见，三角域上的 n 次伯恩斯坦基函数共包含了 $\frac{1}{2}(n+1)(n+2)$ 个基函数. 其个数称为三角数，它等于 n 阶方阵的下三角中所含元素个数，例如 $n=2$ 与 $n=3$ 时分别如图 2-23、图 2-24 所示，其中位于同一条线上的那些基函数实际都是单变量的.

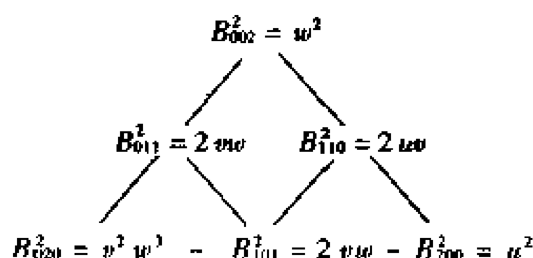


图 2-23

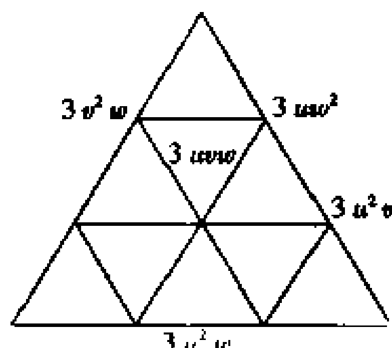


图 2-24

三角域按伯恩斯坦基函数的三角阵列相应划分成子三角域，其中诸直线交点称为节点. 节点与基函数一一对应. 每个节点也用三个指标确定. 它们分别与三参数 $u、v、w$ 相联系. 例如 $n=3$ 时，下标号如图 2-25 所示.

三角域上的伯恩斯坦基函数同样具有规范性、非负性与递推性. 其递推关系为

$$B^n_{i,j,k}(u,v,w) = uB^{n-1}_{i-1,j,k}(u,v,w) + vB^{n-1}_{i,j-1,k}(u,v,w) + wB^{n-1}_{i,j,k-1}(u,v,w).$$

2.7.3 三边贝齐尔曲面片的方程

一张 n 次三边贝齐尔曲面片由构成三角阵列的 $\frac{1}{2}(n+1)(n+2)$ 个控制顶点 $\boldsymbol{b}_{i,j,k}$，$i+j+k=n$，$i、j、k \geqslant 0$ 定义. 曲面片的方程

$$\boldsymbol{B}^n(u,v,w) = \sum_{i+j+k=n} \boldsymbol{b}_{i,j,k} B^n_{i,j,k}(u,v,w), \quad T: 0 \leqslant u,v,w \leqslant 1. \quad (2\text{-}34)$$

按下标顺序用直线连接控制顶点，就形成了曲面的控制网格，它由三角形组成，网格顶点与三角域的节点一一对应. 图 2-26 给出 3 次三边贝齐尔曲面片的一个例子.

当固定三参数之一时，将得到曲面片上一条等参数线. 例如，当 w 固定，让 u 独立地变化，则得一条 u 线；若让 v 独立地变化，则得 v 线，两者实际是同一条曲线. 因此，曲面上有三族等参数线. 当三参数之一为零时，则得曲面片的一条边界曲线，它由相应那排边界顶点定义，就是非有理的 n 次贝齐尔曲线. 当三参数之一等于 1 时，则得三边曲面片的一个角点，就是控制网格三角顶点之一. 可见，三边贝齐尔曲面片有与四边贝齐尔曲面片类似的性质.

与定义在矩形域上的四边贝齐尔曲面片的差别在于：① 定义域不同；② 控制网格不同，后者由呈矩形阵列的控制顶点构成；③ 同样是两个独立的参数，但最高

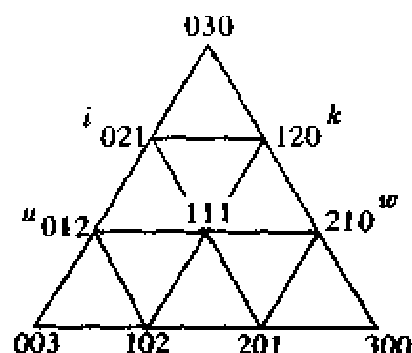


图 2-25

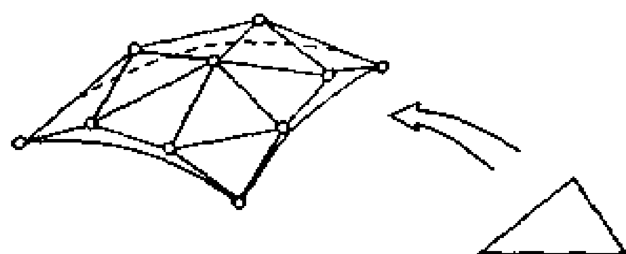


图 2-26

次数不同，后者两个参数的最高次数是互相独立的，可以不同，而三边贝齐尔曲面片的三个参数的最高次数都是一致的；④ 四边曲面片是张量积曲面，三边贝齐尔曲面片是非张量积曲面，这是本质差别.

2.7.4　德卡斯特里奥算法

任意给定 $P=(u,v,w)\in T$，现在用几何作图法来计算三角曲面上的对应点.

给定曲面的控制点 $b_{i,j,k}(i+j+k=n)$，令

$$b_{i,j,k}^{(0)}=b_{i,j,k}\quad(i+j+k=n).\tag{2-35}$$

再定义

$$b_{i,j,k}^{(l)}=ub_{i+1,j,k}^{(l-1)}+vb_{i,j+1,k}^{(l-1)}+wb_{i,j,k+1}^{(l-1)},\tag{2-36}$$

$$l=1,2,\cdots,n,i+j+k+l=n,$$

$$B^n(u,v,w)=b_{0,0,0}^{(n)}.$$

由(2-34)与(2-35)式所定义的算法称为德卡斯特里奥算法，简称递推算法，如图 2-27. 由递推算法最后得到的那个点 $b_{0,0,0}^{(n)}$ 正是三角曲面上对应于 $p=(u,v,w)$ 的点；三点 $b_{1,0,0}^{(n-1)}$、$b_{0,1,0}^{(n-1)}$、$b_{0,0,1}^{(n-1)}$ 所张成的平面是曲面在点 $b_{0,0,0}^{(n)}$ 处的切平面.

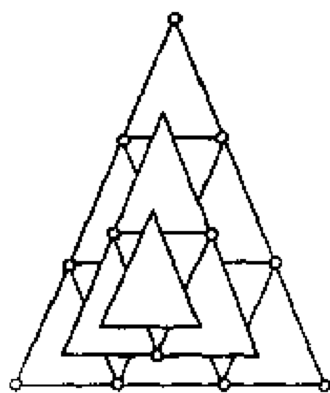


图 2-27

2.7.5　曲面片的分割

给定参数 u、v、w，由德卡斯特里奥算法求出曲面片上一点，同时可得到该点对曲面片进行分割的数据. 三参数决定了三角域内一点，该点与三顶点的连线把三角域划分为三个子三角域. 这三个子三角域所映射的三个子曲面片上的控制顶点已由执行德卡斯特里奥算法求曲面片该点的同时得到，如图 2-28 所示.

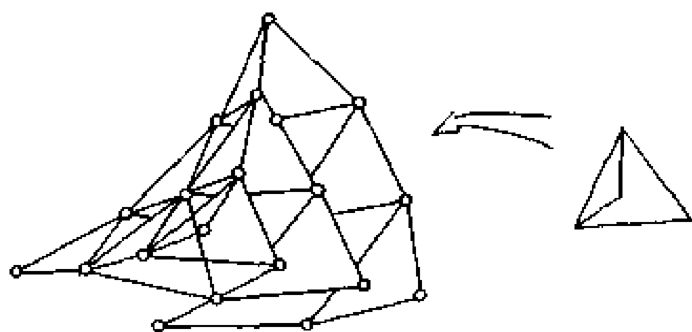


图 2-28

2.7.6　升阶公式

任何一张 n 次贝齐尔三角曲面，总可以形式地表为 $n+1$ 次三角曲面，在(2-34)式中，伯恩斯坦系数是 $\{b_{i,j,k}\}$，$i+j+k=n$. 把(2-34)式右端末乘以 $(u+v+w)$ 得到

$$B^n(P)=\sum_{i+j+k=n+1} b^*_{i,j,k}B^{n+1}_{i,j,k}(P),$$

其中新的伯恩斯坦系数为

$b^*_{i,j,k}=\frac{1}{n+1}(ib_{i-1,j,k}+jb_{i,j-1,k}+kb_{i,j,k-1})$，$i+j+k=n+1$. 当上式的右方的某一项出现负的下标时，这一项应当理解为零.

新的伯恩斯坦系数确定了 T 上的一张 B-网，称为升阶 B-网，它与原来的 B-网定义着一张三角曲面，一般来说，升阶 B-网更加贴近三角曲面；升阶之后，控制点增多了，增加了修改曲面的灵活性.

升阶过程可以一次又一次地进行下去，由此得到一个升阶 B-网序列.

定理 4　当升阶的过程无限制地进行下去时，升阶 B-网序列在 T 上收敛到它们所定义的贝齐尔三角曲面片.

2.8　常用参数曲线、曲面的等价表示

参数曲线曲面的不同表示，在于所选函数空间中的基函数的不同，从而使得所表示的曲线曲面具有不同的几何特性. 但是它们都是参数多项式(样条)曲线曲面，因此相互之间可以等价表示. 这一点有时便于计算几何量，例如幂基形式适合于求导的计算等.

2.8.1　常用参数曲线的等价表示

前面介绍了三次埃尔米特(即 Ferguson)曲线、贝齐尔曲线和 B 样条曲线. 在实际应用中常需要对这三种非有理参数表示式进行相互转换，即对于同一条曲线，已

知这三种表示的一种形式,可以推导出其它两种形式.这三种参数曲线的表示形式可统一写成 $P(t) = TMB$ 的形式,其中,$T = (t^3, t^2, t, 1)$,并令三次埃尔米特曲线为

$$P_H(t) = TM_HB_H,$$

$$M_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, B_H = (P_0, P_1, P'_0, P'_1)^{\mathrm{T}}$$

三次贝齐尔曲线为

$$P_Z(t) = TMZB,$$

$$M_Z = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad B_Z = (P_i, P_{i+1}, P_{i+2}, P_{i+3})^{\mathrm{T}}$$

三次均匀 B 样条曲线为

$$P_B(t) = TM_BB_B,$$

$$M_B = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, B_B = B_Z$$

(1) 已知 M_H, B_H,求 B_Z, B_B, $B_Z = M_H^{-1}M_HB_H$, $B_B = M_B^{-1}M_HB_H$.

(2) 已知 M_Z, B_Z,求 B_H, B_B, $B_H = M_Z^{-1}M_ZB_Z$, $B_B = M_B^{-1}M_ZB_Z$.

(3) 已知 M_B, B_B;求 B_H, B_Z, $B_H = M_H^{-1}M_BB_B$, $B_Z = M_Z^{-1}M_BB_B$.

上述三种情况中,所用到的 M_H^{-1}、M_Z^{-1}、M_B^{-1} 存在,故上式成立.

2.8.2 常用双三次参数曲面的等价表示

一个 $n \times m$ 次参数曲面 $p(u, v)$ 可有多种表示形式,例如幂基形式、贝齐尔形式、孔斯形式、B 样条形式等,它们可看作是在不同的基底 $\{u^iv^j\}$, $\{B_{i,n}(u)B_{j,m}(v)\}$, $\{F_{i,n}(u)F_{j,m}(v)\}$ 下的不同表示.

已知双三次埃尔米特弗格森、贝齐尔、(均匀)B 样条的矩阵表达式是

$$S_H(u, v) = UM_HB_HM_H^{\mathrm{T}}W^{\mathrm{T}},$$

$$S_Z(u, v) = UM_ZB_ZM_Z^{\mathrm{T}}W^{\mathrm{T}},$$

$$S_B(u, v) = UM_BB_BM_B^{\mathrm{T}}W^{\mathrm{T}},$$

其中 M_H, M_Z, M_B 是上一节参数曲线情形.对于一张曲面,若已知上述三种表示形式的一种,如何求得其它两种.

(1) 已知贝齐尔表示,求埃尔米特和 B 样条表示的 B_H 和 B_B:

对同一张曲面有:$M_HB_HM_H^{\mathrm{T}} = M_ZB_ZM_Z^{\mathrm{T}}$, $M_BB_BM_B^{\mathrm{T}} = M_ZB_ZM_Z^{\mathrm{T}}$,

则 $$B_H = M_H^{-1}M_ZB_ZM_Z^{\mathrm{T}}(M_H^{-1})^{\mathrm{T}}, B_B = M_B^{-1}M_ZB_ZM_Z^{\mathrm{T}}(M_B^{-1})^{\mathrm{T}}.$$

(2) 已知埃尔米特表示,求贝齐尔和B样条表示的 $\boldsymbol{B}_H$ 和 $\boldsymbol{B}_B$:

$$\boldsymbol{B}_Z = \boldsymbol{M}_Z^{-1}\boldsymbol{M}_H\boldsymbol{B}_H\boldsymbol{M}_H^{\mathrm{T}}(\boldsymbol{M}_Z^{-1})^{\mathrm{T}},\boldsymbol{B}_B = \boldsymbol{M}_B^{-1}\boldsymbol{M}_H\boldsymbol{B}_H\boldsymbol{M}_H^{\mathrm{T}}(\boldsymbol{M}_B^{-1})^{\mathrm{T}}.$$

(3) 已知B样条表示,求贝齐尔和埃尔米特表示的 $\boldsymbol{B}_Z$ 和 $\boldsymbol{B}_H$.

$$\boldsymbol{B}_H = \boldsymbol{M}_Z^{-1}\boldsymbol{M}_B\boldsymbol{B}_B\boldsymbol{M}_B^{\mathrm{T}}(\boldsymbol{M}_Z^{-1})^{\mathrm{T}},\boldsymbol{B}_H = \boldsymbol{M}_H^{-1}\boldsymbol{M}_B\boldsymbol{B}_B\boldsymbol{M}_B^{\mathrm{T}}(\boldsymbol{M}_H^{-1})^{\mathrm{T}}.$$

2.9 扫描曲面

2.9.1 扫描曲面的定义

扫描(sweep,或称扫掠)曲面生成的基本思想是曲线运动生成曲面.这一造型方法在曲面生成和设计中方便灵活,应用广泛.

决定扫描曲面生成的因素有三个:截线、脊线(或称路径)和扫描规则.截线决定曲面的横截面形状,脊线决定曲面的纵向走向,而扫描规则决定截线运动时在脊线上的定位方式以及截线运动时的连续形变方式.因此,扫描曲面可看作初始截线作刚体运动和局部连续形变的合成而生成的曲面.截线的运动方式称为扫描运动(或扫掠运动).

扫描面的生成方法灵活多样,按截线沿脊线的运动方式分类,有平移、旋转、环行、垂直、同步等多种形式;按截线、脊线的数目分类,有单截线单脊线、单截线多脊线、多截线单脊线、多截线多脊线等.

目前,扫描面的构造方式主要有以下几种:

(1) 运动和变换的矩阵表示.截面线在脊线各位置上的形状用矩阵的乘积形式表示.

(2) 局部标架方式.脊线上每一点确定一个局部标架.截线在脊线上的定位和形变用局部标架来表示.这种方式简单、直观,截线定位较易掌握,采用的标架有:弗朗内标架、最小旋转标架及广义平移标架等.

(3) 等距线(offset)方式.将截线的形变和定位看作是单位圆周的变半径等距曲线,或者把曲面的纵向轮廓线看作是脊线的变半径等距线.

以上三种方式都可以用表达式精确表示扫描面.有的造型系统把扫描体作为CSG的基本体素,这种体素表示的主要优点之一是减少了体素和布尔运算的数目,缩短了CSG的深度.由于扫描面(体)的生成特点,造型中常用的两种体素:拉伸体和回转体,都可以看作是扫描体的特例,但一般扫描体的求交和显示都有其自身特点.另外造型系统的数据结构也会随加入扫描体素而有所改变,也有许多造型系统并不把一般扫描体作为基本体素.因此,对于基于边界表示的造型系统,扫描体需要作从精确表示到边界表示的近似转换.常用的转换方式是用上述(2)、(3)两种方式生成一族边界曲线,再用蒙皮(Skinning)方法表示张量积曲面.

2.9.2 扫描面的表示

1.运动表示

$$\boldsymbol{S}(u,s) = \boldsymbol{\xi}(s) + \boldsymbol{B}(s)\boldsymbol{f}^{q}(u) + \boldsymbol{H}(\boldsymbol{f}^{q}(u),s), \tag{2-37}$$

式中 $\xi:[0,1]\to \mathbf{R}^3, \boldsymbol{B}:[0,1]\to GL^+(3)$ 都是光滑映照，且 $\xi(0)=0, \boldsymbol{B}(0)=\boldsymbol{I}$，以及 $\det(\boldsymbol{B})>0$，$\xi(s)$ 表示脊线，$f^0(u)$ 表示初始截线，$\boldsymbol{H}(f^0(u),s)$ 是曲线 $f^0(u)$ 的连续形变的非线性项.

当 u 取遍 $[u_1,u_2]$，s 取遍 $[0,l]$ 时，上式表示由初始截线在时刻 $[0,l]$ 内运动生成的扫描曲面.

2. 局部标架表示

设沿 $\xi(s)$ 取局部坐标系为 $E\{\xi(s);\boldsymbol{e}_1(s),\boldsymbol{e}_2(s),\boldsymbol{e}_3(s)\}$，设 $\xi(s)$ 处的截线记为 $f(u,s)$，在此局部坐标系下的表示为 $(f_1(u,s),f_2(u,s),f_3(u,s))$，则

$$\boldsymbol{S}(u,s)=\xi(s)+f_1(u,s)\boldsymbol{e}_1+f_2(u,s)\boldsymbol{e}_2+f_3(u,s)\boldsymbol{e}_3 \tag{2-38}$$

(2-37) 式表示扫描曲面的运动表示形式，(2-38) 式表示扫描曲面在局部标架下的表示形式. 下式是两种表示之间的关系：

$$f(u,s)=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}(s)f^0(u)+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{H}(f^0(u),s) \tag{2-39}$$

式中 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}$ 表示在局部标架下的仿射阵，$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{H}(f^0,s)$ 表示在局部标架下的形变的非线性项，$\boldsymbol{A}$ 表示 $\xi(s)$ 上的局部标架到世界坐标系下的坐标变换，所以上式表示初始截线在局部标架上的零点固定的一般形变，也表示(2-37)、(2-38) 两种表示之间的关系式.

下面是一些特定的扫掠运动：

(1) 若扫掠运动是线性形变，即(2-39) 中 $\boldsymbol{H}=0, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}=\boldsymbol{D}(s)$，$\boldsymbol{D}(s)$ 表示只在局部标架 E 的三个坐标轴方向的伸缩形变阵，则(2-38) 式中写成，

$$\boldsymbol{S}(u,s)=\xi(s)+f_1^0(u)d_1(s)\boldsymbol{e}_1+f_2^0(u)d_2(s)\boldsymbol{e}_2+f_3^0(u)d_3(s)\boldsymbol{e}_3, \tag{2-40}$$

它表示(2-38) 式中的 $f_i(u,s)$ 可写成两个函数之乘积形式($i=1,2,3$)

(2) 若初始截线是平面截线，且连续形变是平面仿射形变，截线面取为 $(\boldsymbol{e}_2,\boldsymbol{e}_3)$ 坐标平面，则(2-40) 式可写成：

$$\boldsymbol{S}(u,s)=\xi(s)+f_2(u,s)\boldsymbol{e}_2+f_3(u,s)\boldsymbol{e}_3. \tag{2-41}$$

截线的连续形变通常由形变曲线来限定：① 多脊线情形. 在相应脊线位置限定截线的形变方式，② 多截线情形. 利用两截线之间的形变(简单情形是取线性形变) 插值多截线，或者不用脊线(限于多截线情形，此时为蒙皮方法). 形变方式的确定是扫描面设计中最灵活的手段之一.

(2-38) 式中的局部标架的确定灵活多样，脊线上任一点可以定义多种局部标架形式，下列几种方式较为常用：

(1) 弗朗内标架　是最常用的一种，但要求脊线是 C^2 连续，而且沿曲线移动时易引起标架位置的突变，定位不易掌握. 改进的弗朗内标架可解决标架突变问题.

(2) 最小旋转标架　目的是为了消除脊线上局部标架的不必要的旋转，它是通过一个微分方程组来定义的：

$$\begin{cases} \boldsymbol{q}'(s)=-\langle \boldsymbol{r}''(s),\boldsymbol{q}(s)\rangle \boldsymbol{r}'(s)/|\boldsymbol{r}'(s)|^2, \\ \boldsymbol{m}'(s)=-\langle \boldsymbol{r}''(s),\boldsymbol{m}(s)\rangle \boldsymbol{r}'(s)/|\boldsymbol{r}'(s)|^2. \end{cases} \tag{2-42}$$

若选取初值为 $\boldsymbol{q}(0)=\boldsymbol{q}_0, \boldsymbol{m}(0)=\boldsymbol{m}_0$，使得 $\boldsymbol{r}'(0)$、$\boldsymbol{q}_0$、$\boldsymbol{m}_0$ 是三个相互垂直的单位

向量,则在脊线上任一点,由(2-42)式满足初始值的解 $q(s)$、$m(s)$ 和单位切向 $r(s)/\!/|r'(s)|$ 一起构成脊线上一点的正交标架.这里 $< r'', q >$ 表示两矢量的内积.

容易找出上述两种标架之间的转换关系.也可以利用弗朗内标架和这种转换关系,直接得到最小旋转标架的表示形式.最小旋转标架有如下性质:

当且仅当脊线是平面曲线,且曲线上无曲率为0的点时,对于3D脊线和2D截线,选取这两种标架生成的扫描面是一致的.

(3) 广义平移标架　这种标架比前述两种方法更易计算,对脊线的连续性要求仅是 G^1 连续.设脊线为 $r = r(s) = (r_1(s), r_2(s), r_3(s))^T, s \in [0,1)$ 是连续的空间曲线,为方便计,设脊线是一阶光滑曲线,且至少在一点处是有二阶导数,不妨设这样的点是脊线的初始点 $r = r(0)$,则在 $r = r(0)$ 处可定义弗朗内标架 $\{r(0); T^0, N^0, B^0\}$.在脊线的任一点 $r = r(s)$ 处,记这一点单位切向为 t,并定义 $n = c_1(s)B^0 \times t, b = t \times n$,式中 $c_1(s) = |B^0 \times t|^{-1}$,$(t, n, b)$ 成正交标架.显然有

$$\begin{cases} t = t, \\ n = c_1(s) \cdot B^0 \times t, \\ b = c_1(s)(B^0 - \langle B^0, t\rangle t), \end{cases}$$

式中 $c_1(s) = |B^0 \times t|^{-1}$,这就是定义在 $r = r(s)$ 上任一点处的一个广义平移标架,n、b 在脊线 $r = r(s)$ 一点的法平面上.由标架的定义可知,n 总是平行于初始标架 T^0、N^0 张成的平面.这里 B^0 也可以取作某一固定矢量.例如取世界坐标系中的一坐标方向等.

上述标架沿脊线运动时,关于初始切向量没有旋转.下面是这一标架与弗朗内标架的关系:

对于3D脊线和2D截线,选取弗朗内标架和广义平移标架生成的扫描面是一致的,当且仅当脊线是定倾曲线.

造型中常用的方法是用上述标架在脊线上各点定位截线,再用蒙皮方法表示扫描曲面.利用局部标架,容易实现对截线作扭转形变等操作.

2.10　基于三维散乱数据的曲面拟合

2.10.1　曲面的构造

在地质勘探、气象预报等领域,常会遇到利用二元函数 $F(x,y)$ 对平面区域 Ω 上随机分布的散乱点 (x_i, y_i) 处的观测值 $f_i, i = 1,2,\cdots,N$,进行曲面拟合的问题,即求区域 Ω 上的函数 $F(x,y)$,使得

$$F(x_i, y_i) = f_i, i = 1,2,\cdots,N,$$

或者求 $F(x,y)$,使得

$$\min \sum_{i=1}^{N} (F(x_i, y_i) - f_i)^2.$$

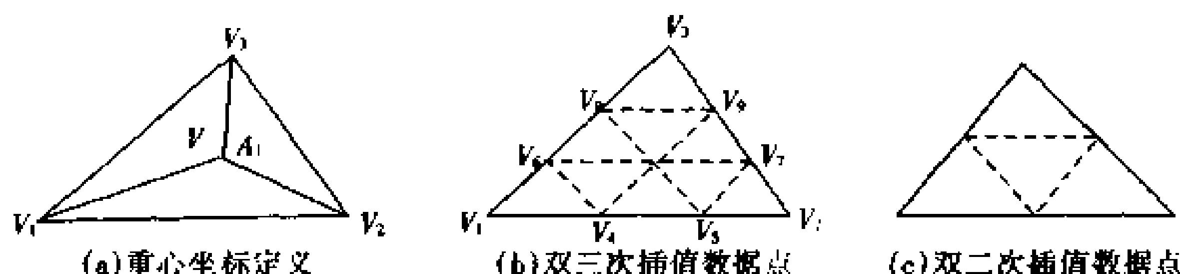
如果能确定 $F(x,y)$ 所在的函数空间，并能形式地表示出来，那么用最小二乘法求解是很方便的. 比如把区域 Ω 用矩形网格分割(或用矩形网格覆盖)，将 $F(x,y)$ 取为乘积型B样条函数，利用最小二乘法反求出B样条函数的控制顶点. 但有时最小二乘法不能够满足要求. 有些问题是需要所求函数对原始数据进行插值的，在这种情况下，利用各种剖分下的曲面插值格式进行曲面拟合是必要的.

3D散乱数据的光滑插值一般可叙述为：已给一组3D数据 (x_i,y_i,z_i)，$i=1,2,\cdots,N$，构造一个具有所需光滑度的曲面 $z=S(x,y)$，使之插值于这些数据，即满足 $S(x_i,y_i)=z_i$，$i=1,2,\cdots,N$，对数据点 (x_i,y_i) 要求互不重合且不共线.

对于较规则的四边形网格分布的数据，可以利用张量积曲面插值，问题归结为曲面的反算问题，但许多情形下，给定二三维数据是不规则的，在三角域上构造曲面更为适宜. 最常见的曲面拟合是，在三角剖分下，利用二元样条函数进行光滑插值. 用样条函数进行曲面拟合的优点是次数较低，局部性好，比如孔斯曲面、Powell-Sabin格式、Hsieh Clough-Tocher格式等. 由于散乱数据的曲面拟合对光滑性要求不是很高，一般为一阶、二阶光滑，因此在实际应用中，普遍采用低次样条曲面，常见的有双三次孔斯曲面、二次 C^1Powell-Sabin格式、三次 C^1HCT格式、三角域上的超限插值格式等.

下面导出分片三角曲面的拉格朗日插值公式.

设插值曲面 S 定义在 xy 平面上区域 Ω 中，$(x_i,y_i)\in\Omega$，$i=1,2,\cdots,N$. 将 $V_i=(x_i,y_i)$ 作为顶点，对区域 Ω 作三角剖分，在每个三角形上求插值函数 $S(x,y)$. 希望插值曲面具有 C^1 连续性，即它的切平面连续变化. 对 xy 平面上任意三角形 T，顶点为 V_1、V_2、V_3、V 是平面上任一点图2-29(a)，可以定义 V 相对于 T 的重心坐标 (u_1,u_2,u_3)，于是 $V=\sum_{i=1}^{3}u_iV_i$. 又因一个双三次多项式有10个未知系数，故需要10个插值条件才能确定. 若在 T 上已知图2-29(b)所示的10个标记点的数据，即 $z_i=S(u_i)$，$i=1,2,\cdots,10$，则可以确定插值曲面 $S(x,y)$ 为



(a)重心坐标定义　(b)双三次插值数据点　(c)双二次插值数据点

图 2-29

$$S(x,y)=\sum_{i=1}^{10}z_i\varphi_i(x,y),$$

其中 $\varphi_i(x,y)$ 属于双三次多项式函数类，且满足 $\varphi_i(V_j)=\delta_{ij}$，$i,j=1,2,\cdots,10$. 据此不难导出用重心坐标表示的双三次拉格朗日插值函数

$$\varphi_1 = \frac{9}{2}u_1(u_1-\frac{1}{3})(u_1-\frac{2}{3}),\quad \varphi_2 = \frac{9}{2}u_2(u_2-\frac{1}{3})(u_2-\frac{2}{3}),$$
$$\varphi_3 = \frac{9}{2}u_3(u_3-\frac{1}{3})(u_3-\frac{2}{3}),\quad \varphi_4 = \frac{27}{2}u_1u_2(u_1-\frac{1}{3}),$$
$$\varphi_5 = \frac{27}{2}u_1u_2(u_2-\frac{1}{3}),\quad \varphi_6 = \frac{27}{2}u_1u_3(u_1-\frac{1}{3}),$$
$$\varphi_7 = \frac{27}{2}u_2u_3(u_2-\frac{1}{3}),\quad \varphi_8 = \frac{27}{2}u_1u_3(u_3-\frac{1}{3}),$$
$$\varphi_9 = \frac{27}{2}u_2u_3(u_3-\frac{1}{3}),\quad \varphi_{10} = 27u_1u_2u_3.$$

可以证明上式具有仿射不变性，且具有 C^1 连续性，适合用于三维图形的表示.

类似地，对双二次插值曲面有(6个未知系数)，若已知 T 上数据点 $z_i = S(p_i)$，$i = 1,\cdots,6$，如图 2-33(c) 所示，则有插值曲面的重心坐标式

$$S(x,y) = \sum_{i=1}^{6} z_iQ_i(x,y),$$

其中，$Q_1 = u_1(2u_1-1)$，$Q_2 = u_2(2u_2-1)$，$Q_3 = u_3(2u_3-1)$，$Q_4 = 4u_2u_3$，$Q_5 = 4u_1u_3$，$Q_6 = 4u_1u_2$，双二次拉格朗日插值在分片三角曲面上是 C^1 连续的，但在分片曲面连接处可能有切向跳跃，达不到全局 C^1 连续.

利用拉格朗日插值不需要插值点处的微商信息，但这一方法不具有局部性质.利用样条函数的插值格式，虽然具有许多优点，但需要在插值点处计算或给定一阶或二阶偏导数等几何插值信息.

利用样条函数进行散乱数据曲面插值通常需要三个步骤：① 形成三角剖分，即以散乱点为剖分网点，把定义域剖分成若干个三角形区域；② 偏微商信息的计算；③ 构造插值样条曲面.

2.10.2　三角剖分及其优化

已知平面区域 Ω 上的散乱点 $V_i = (x_i, y_i)$，$i = 1,2,\cdots,N$，其中 Ω 的边界是由某些散乱点连成的多边形，如果下面的网点集合(或顶点集合)

$$V = \{(x_i,y_i) \mid i = 1,2,\cdots,N\},$$

网线集合(或边的集合)

$$E = \{e_{ij}(V_i,V_j) \mid i \neq j, V_i, V_j \in V \text{ 且相邻}\},$$

以及网面集合(或三角形集合)

$$T = \{t_{ijk}(V_i,V_j,V_k) \mid V_i、V_j、V_k \text{ 不共线，且} (V_i,V_j),(V_j,V_k),(V_k,V_i) \in E\}$$

构成的并集 Δ 满足条件：

1° 如果一个三角形属于 Δ，则它的所有边和顶点都属于 Δ；

2° 如果两个 $i(i = 1,2)$ 维单形 S_1、$S_2 \in \Delta$，那么 $S_2 \cap S_2$ 要么是空集，要么是它们的公共边或顶点，且是 Δ 中小于 i 维的单形；

3° $\bigcup_{t_a \in T} t_a = \Omega$；

则称 Δ 为 Ω 上的关于散乱点 $V_i, i = 1,2,\cdots,N$，的三角剖分，若 $e_{ij} \in E$ 是 T 中两个三角形的公共边，则称 e_{ij} 为内网线，否则称为边界网线．边界网线的顶点称为边界网点．

对于散乱数据，先形成三角剖分，然后再应用三角剖分下的曲面格式进行曲面插值或逼近．形成三角剖分实际上是构造 C^0 插值．但对于大多数散乱数据来讲，相应的三角剖分不是唯一的．因此，应根据实际需要在诸多中找出最优的剖分，这一过程叫做三剖部分的优化．现在，关于三角剖分及其优化的方法很多，下面是一些剖分优化的方法．

对于任意内网线 $e_{ij} \in E$，以 e_{ij} 为公共边的两个三角形记为 $t_{ijk}(V_i, V_j, V_k)$ 和 $t_{ijl}(V_i, V_j, V_l)$．如果四边形 (V_i, V_k, V_j, V_l) 为凸四边形，且 V_i、V_k、V_j、V_l 中任意三点不共线，则四边形 (V_i, V_k, V_j, V_l) 存在另一种三角部分（如图 2-30）．令 t^*_{ikl}、t^*_{jkl} 表示第二种剖分对应的两个三角形，相应的新的内网线记为 e^*_{kl}．现在要问，在以上两种剖分中，应如何选出其中最合适的剖分呢？为此，引入衡量以上凸四边形之三角剖分的正则性的参量 s，如果 $s(t_{ijk}, t_{ijl}) < s(t^*_{ikl}, t^*_{jkl})$，则说新剖分 (t^*_{ikl}, t^*_{jkl}) 的正则性比旧剖分 (t_{ijk}, t_{ijl}) 的正则性好，从而应该采用新剖分 (t^*_{ikl}, t^*_{jkl})，正则性 s 有不同的定义方法，常见的有

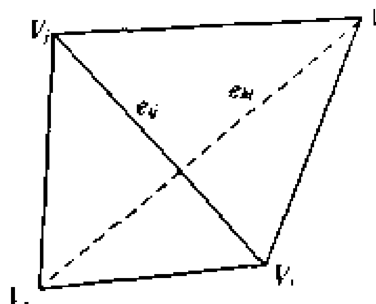


图 2-30

(1) $s(t_{ijk}, t_{ijl})$ 等于 t_{ijk} 和 t_{ijl} 的最小内角；

(2) $s(t_{ijk}, t_{ijl}) = \min\left\{\dfrac{A_{ijk}}{L^2_{ijk}}, \dfrac{A_{ijl}}{L^2_{ijl}}\right\}$，其中 A_{ijk} 为三角形 t_{ijk} 面积，L_{ijk} 为 t_{ijk} 的周长；

(3) $s(t_{ijk}, t_{ijl}) = \left\{\dfrac{r_{ijk}}{R_{ijk}}, \dfrac{r_{ijl}}{R_{ijl}}\right\}$，其中 r_{ijk} 为三角形 t_{ijk} 的内切圆半径，R_{ijk} 为外接圆半径．

总之，三角剖分优化的原则是使每个小三角形尽可能"臃"，避免出现狭长的三角形．剖分优化的过程是所有内网线的相邻两个三角形都要拿优化准则(1)、(2)或(3)来判断是否修改，直至所有内网线不能修改为止，剖分达到最优．

对于三维空间中的任意散乱点进行三角剖分，目前尚无较好的方法．

2.10.3 型值点处微商信息的估值

许多曲面拟合方法都需要估计网点处的微商信息，它们将直接影响最终曲面的形状，所以如何处理好这一问题，使得满足一定的要求（例如，保真度）是很重要的．

在三角剖分下，网点处微商信息的估值方法如下：

(1) 凸组合方法　实际上这种方法是通过与某一点相邻的三角形或全体三角形（空间）所在平面的法矢（或偏微商）做凸组合来近似替代该点的法矢（或偏微商）．例如

$$S_x(v_i)=\frac{\sum\limits_{(i,j,k)\in N_i}W_{ijk}f_x(i,j,k)}{\sum\limits_{(i,j,k)\in N_i}W_{ijk}},S_y(v_i)=\frac{\sum\limits_{(i,j,k)\in N_i}W_{ijk}f_y(i,j,k)}{\sum\limits_{(i,j,k)\in N_i}W_{ijk}},$$

其中 N_i 和 W_{ijk} 有不同的给法．例如 Akima 方法，$N_i\{(i,j,k)\mid j\neq k,V_j、V_k$ 是 V_i 的邻点$\}$；$W_{ijk}=\cos(\gamma_{ijk})\cdot S_{ijk}$，其中 S_{ijk} 为三角形$\{(V_i,f_i),(V_j,f_j),(V_k,f_k)\}$面积，$\gamma_{ijk}$ 为以上三角形所在平面的法矢与 z 轴的夹角．

(2) 李特尔(Little)方法　$N_i\{(i,j,k)\mid V_i、V_j、V_k$ 构成三角形$\}$，

$$W_{ijk}=\frac{1}{|l_{ij}|^2|l_{ik}|^2},$$

其中 $|l_{ij}|$ 表示 $\overline{V_iV_j}$ 的长度．

(3) 节点函数法　令任意点(x_i,y_i)对应一个结点函数 $Q_i(x,y)$，$Q_i(x,y)$ 为二次函数：

$$Q_i(x,y)=f_i+a_i(x-x_i)+b_i(y-y_i)+c_i(x-x_i)^2+$$
$$d_i(x-x_i)(y-y_i)+e_i(y-y_i)^2.$$

显然 $Q_i(x_i,y_i)=f_i$，通过加权最小平方拟合，可以求得参数 $a_i、b_i、c_i、d_i、e_i$，权取为距离的倒数，最后求得

$$S_x(V_i)=a_i,S_y(V_i)=b_i,\quad S_{xx}(V_i)=2c_i,\quad S_{yy}(V_i)=2e_i,\quad S_{xy}(V_i)=d_i.$$

(4) 最小能量法　存在唯一的函数 f 使得 $\int_E\left[\frac{\partial^2 f}{\partial s^2}\right]^2\mathrm{d}s$ 最小．其中 E 为三角剖分中所有三角形边的集合．由此可以得到 $f_x、f_y$ 即为所估计值的偏导数值．

或者是通过解 $\int_Q(f_x^2+f_y^2)\mathrm{d}s$ 最小而得到 $f_x、f_y$ 的值．

(5) 方程求解法　一种全局性的估值方法，这种方法行之有效，处理方法同三次样条的累加弦长的连续性方程组．

对于空间三角剖分 T，设 M_i 表示顶点 v_i 邻点的集合；T_i 表示以 V_i 为顶点的空间三角形集合；k_i 为 M_i 的维数；$\boldsymbol{n}_i$ 代表 V_i 处曲面的法矢；N_i 为三角形 $t_i\in T_i$ 的法矢，则有下式：

$$k_i\boldsymbol{n}_i+\sum_{V_j\in M_i}a_{ij}\boldsymbol{n}_j=(k_i+1)\sum_{t_j\in T_i}\beta_{ij}N_i,\quad i=1,2,\cdots,N,$$

其中

$$a_{ij}=\frac{|\overline{V_iV_j}|}{\sum\limits_{v_k\in M_i}|\overline{V_iV_k}|};\quad \beta_{ij}=\frac{s(t_j)}{\sum\limits_{t_j\in T_i}s(t_j)},S(t_j)\text{—三角形 } t_j \text{ 的面积}$$

以上将产生一个 N 阶的对角占优的方程组，它的解就是所求的 $\boldsymbol{n}_i$．

2.10.4 几种曲面拟合方法

1. Shepard 方法

针对散乱分布的数据，Shepard 提出了如下的有理插值曲面：

$$F(x,y)=\begin{cases}\sum_{i=0}^{N}\omega(r_i)f_i\Big/\sum_{i=1}^{N}\omega(r_i), & \\ & r_i\neq 0\\ f_i & r_i=0\end{cases}\tag{2-43}$$

其中 $\omega(r_i)=1/r_i^{\alpha},\alpha>0,r_i=[(x-x_i)^2+(y-y_i)^2]^{1/2}$.

曲面(2-43)是关于插值数据(x_i,y_i,f_i)的全局插值曲面.当(x,y)不等于$(x_i,y_i),i=1,2,\cdots,N$时,$F(x,y)$是所有函数值$f_i(i=1,2,\cdots,n)$的加权平均,权因子$\omega(r_i)$与(x,y)到各插值点距离有关,f_i的权系数可写成

$$W_i(x,y)=\omega(r_i)\Big/\sum_{i=1}^{N}\omega(r_i)=\prod_{\substack{j=1\\ j\neq i}}^{N}r_j^{\alpha}\Big/\sum_{i=1}^{N}\prod_{\substack{j=1\\ j\neq i}}^{N}r_j^{\alpha},$$

$$j=1,\cdots,N,0<\alpha+\infty,$$

它满足

$$W_i(x_i,y_i)=\delta_{ij};i,j=1,2,\cdots,N.$$

Shepard曲面的缺点是,它是全局的且代数精度低,只有零次代数精度;其优点是算法简单,曲面光滑,插值灵活.为了克服全局性的缺点,Shepard本人提出一个局部逼近方案.适当选定$R>0$,并取权因子

$$\omega(r_i)=\begin{cases}1/r_i, & 0<r_i\leqslant\dfrac{R}{3},\\ \dfrac{27}{4R}\left(\dfrac{r_i}{R}-1\right)^2, & \dfrac{R}{3}<r_i\leqslant R,\\ 0, & r_i>R,\end{cases}$$

这个权因子是连续可微的,并且支集是有界的.

如果给出在插值点(x_i,y_i)处的一阶偏导数f_{x_i}、f_{y_i}那么插值曲面

$$F(x,y)=\sum_{i=1}^{N}W_i(x,y)[f_i+(x-x_1)f_{x_i}+(y-y_i)f_{y_i}],$$

满足

$$F(x_i,y_i)=f_i,\quad \frac{\partial}{\partial x}F(x_i,y_i)=f_{x_i},\quad \frac{\partial}{\partial y}F(x_i,y_i)=f_{y_i}.$$

2. 二步逼近法

为了构造散乱数据的拟合曲面,又不需要过多的计算量,L.L.Schumaker提出了曲面拟合的二步逼近法.二步逼近法的主要思想是:第一步分片构造散乱数据的拟合曲面(例如最小二乘曲面),各片之间可以是不连续的.第二步重新规则插值点(x_i,y_i),并通过分片拟合曲面计算出该点的函数值f_i(或偏导值),构造光滑的拟合曲面.二步逼近法用起来可以灵活多变,一般地第一步多采用简单的低次的曲面拟合方法,而第二步则采用较规则的光滑曲面插值格式(如孔斯曲面等).

3. 用双三次B-样条曲面片拟合散乱数据

首先将区域$=[a,b]\times[c,d]$剖分成规则的矩形网格,即根据散乱点(X_k,Y_k)的分布,用直线$x=u_i,y=v_i$将区域Ω划分为若干个小矩形$D_{ij}=[u_{i-1},u_i]\times$

$[v_{j-1},v_j]$, $i=1,2,\cdots,m$; $j=1,2,\cdots,n$. 如果分别取 u_i, v_j 为 x, y 方向的B-样条函数的参数节点，相应的双三次B-样条函数为

$$F(x,y)=\sum_{i=0}^{m}\sum_{j=0}^{n}c_{ij}N_{i,3}(x)N_{j,3}(y),$$

其中 c_{ij} 由最小二乘法求解而得，即由方程组

$$\frac{\partial}{\partial c_{ij}}\left[\sum_{k=1}^{N}(F(x_k,y_k)-f_k)^2\right]=0,$$
$$i=0,1,\cdots,m;j=0,1,\cdots,n$$

唯一确定.

为了曲面逼近性更好，应根据散乱点分布的疏密程度来决定参数节点的疏密，利用非均匀B-样条曲面来逼近散乱数据.

参 考 文 献

1 Farin G. Curves and surfaces for computer aided geometric design--A pactical guide. 3rd ed. Boston: Academic Press. 1993.
2 Piegl, L. &Tiller, W. The nurbs books. Berlin: Springer verlag. 1995.
3 常庚哲. 曲面的数学. 长沙: 湖南教育出版社, 1994.
4 施法中. CAGD&NURBS. 北京: 北京航空航天大学出版社, 1994.
5 苏步青, 华宣积, 应用几何教程. 上海: 复旦大学出版社, 1990.
6 苏步青等. 微分几何教程. 北京: 高等教育出版社, 1979.
7 苏步青, 刘鼎元. 计算几何. 上海: 上海科技出版社, 1980.
8 孙家广等. 计算机图形学(第三版). 北京: 清华大学出版社, 1998.
9 周蕴时等. CAGD中的曲线与曲面. 长春: 吉林大学出版社, 1993.

·计算机数学卷·

第21篇

S 计算几何

编　者　周培德
审校者　卢开澄

目　录

引 言

1975 年，在沙莫斯(Shamos Michael Ian)的文章中出现了“计算几何”这一名称，从此计算几何诞生了(为了与其它亦称为“计算几何”的学科不混淆，本篇称为 S 计算几何或者几何算法). 自那时以来该研究领域取得了辉煌的成果，使得 S 计算几何成为理论计算机科学领域中一个新的极有生命力的子领域，并且该子领域中的研究成果已在计算机图形学、化学、统计分析、模式识别、地理数据库以及其它许多领域中得到了广泛的应用.

S 计算几何研究的典型问题由几何基元(geometric primitives)、几何查找、几何优化等问题类组成. 几何基元包括凸壳、沃罗诺(Voronoi)图、多边形的三角剖分、划分问题与相交问题等；几何查找包括点定位、可视化、区域查找等问题，计算机图形学、数据库中的区域查找、地理图形中的点及点集定位等都是几何查找中的典型例子；几何优化包括参数查找和线性规划.

此外，S 计算几何中各种问题的下界的确定，推导下界的方法以及求解各种几何问题算法的复杂性分析等也是 S 计算几何研究的重要内容.

S 计算几何的新近发展包括几何抽样理论，计算实代数几何，计算拓扑，运动规划，并行计算几何，随机几何算法设计与分析，结构和图形，网络生成，计算机视觉中的几何问题，等等.

本篇主要介绍几何查找，多边形及多边形的划分，凸壳，沃罗诺图，几何体的交，几何体的排列，矩形几何，算法的运动规划等.

1 几 何 查 找

几何查找或称几何检索是指属性相同的一批几何对象(比如点、直线段、圆、多边形、多面体等)中定位某个指定的几何对象，或者在某个特定的域中寻找该域所包含的具有某种属性的所有几何对象.

几何查找的耗费包括询问时间(回答一次询问所要的时间)，存储空间(数据结构占用的内存)，预处理时间(组织数据或某种结构的时间)，修改时间(指定几何对象所对应的数据插入数据结构或从数据结构中删去所需要的时间). 询问时间、存储和预处理时间之间可以进行折衷.

1.1 点定位问题

点定位问题(即点 p 位于区域 R 中)与点包含问题(即区域 R 内包含点 p)的含义是相同的. 这类问题的求解主要依赖于空间的划分. 在平面情况下，考虑直线段

构成的平面划分(又称平面剖分),并且假定这种划分所形成的子区域是连通的.下面介绍两个问题及求解它们的算法.

问题1　给定简单多边形 P 及点 q,确定 P 是否包含点 q 或者点 q 是否在 P 内?

设简单多边形 P 的顶点序列为 $p_1, p_2, \cdots, p_n$,求解问题的一种方法是过点 q 作水平射线 l,如果 l 与 P 的边界不相交,则点 q 在 P 的外部.如果 l 和 P 的边界相交,计数交点数并依据交点数的奇偶性可以判定点 q 是否在 P 的内部.具体地说,交点数为奇(偶)数时,点 q 在 P 的内(外)部.由于这里的多边形不一定是凸的,所以上述的判断方法对某些特殊情况是不适合的,要注意区分这些特殊情况,有关这些情况见图1-1.

图1-1中(a)是正常情况,有一个交点,图(b)中有三个交点,这两种情况均表明点在多边形内部.图(c)应该看成有一个交点,图(d)看成没有交点,图(e)和图(f)有多个交点.我们需想办法把这些情况区分开,为此引入两个函数,即 intersect(l_1, l_2) 和 online(l, p).仅当 l_1 和 l_2 相交而且不交在端点上时,intersect(l_1, l_2)才为真.当点 p 在 l 上时,online(l, p)才为真.如果 $p[i]$ 和 $p[i+2]$ 在 l 的两侧,则是图1-1(c)的情况.如果 $p[i]$ 和 $p[i+2]$ 在 l 的同侧,则是图1-1(d)的情况.如果 $p[i]$ 和 $p[i+3]$ 在 l 的两侧(或同侧),则是图1-1(e)(或(f))的情况.

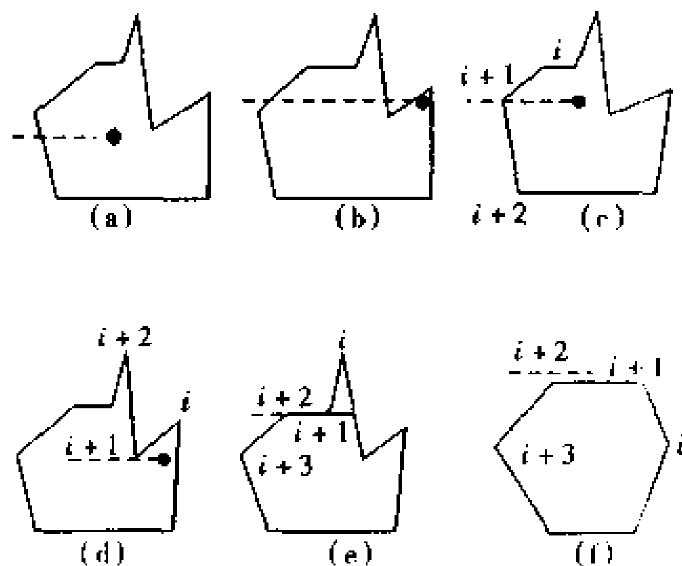


图1-1

下面是计算online函数和判断一个点是否在多边形内部的程序(设多边形有 n 个顶点).

```
Var p:array[1..n + 1]of point;
function online (l:line;p:point):boolean;
Var dx,dy,dx1,dy1:real;
begin
```

```
     dx ← l.p2.x − l.p1.x
     dy ← l.p2.y − l.p1.y
     dx1 ← p.x − l.p1.x
     dy1 ← p.y − l.p1.y
     online ← dx * dy1 − dy * dx1 = 0   and
                (dx1 * (dx1 − dx) < 0 or dy1 * (dy1 − dy) < 0)
end;
function inside (q:point):boolean
Var c,i:integer
     l1,l2:line
begin
       c ← 0
     l1.p1 ← q
     l1.p2 ← q
     l1.p2.x ← maxint
     for  i = 1  to  n  do
       begin
            l2.p1 ← p[i]
            l2.p2 ← p[i + 1]
            if   intersect(l1,l2) or online(l1,p[i + 1]) and [not online(l1,p[i + 2]) and
                 not same 1 (l1,p[i],p[i + 2]) or online (l1,p[i + 2]) and not same 1
                 (l1,p[i],p[i + 3])]
                 then c ← c + 1
       end
       inside ← (c mod 2 < > 0)
end
```

其中函数 same 1 为

$$\text{same } 1 = (dx * dy_1 - dy * dx_1) * (dx * dy_2 - dy * dx_2) \geqslant 0.$$

由于 P 有 n 条边,要检查每条边是否与 l 相交,所以该算法的时间复杂性为 $O(n)$.

现假设 P 是凸 n 边形,点 z 在 P 的内部,从 z 出发作过 P 各顶点的射线,这些射线将平面划分成 n 个楔形(图 1-2),P 的每条边又将相应楔形分为两部分:P 的内部与 P 的外部.以 z 为坐标原点,射线以角次序出现,可以用对分查找判定点 q 所在的楔形,然后再判定 q 位于相应 P 边的哪一侧,便可确定 q 是否在 P 内.这种方法的预处理包括寻找 P 的内点 z,以及组织数据使适合于对分查找.由于事先给定了序列 $p_1,p_2,\cdots,p_n$,所以用 $O(n)$ 时间能建立适于对分查找的二叉树.具体查找 q 是否在 P 内只需要 $O(\text{lb}n)$ 时间.

为了能进行对分查找,P 的顶点一定要关于点 z 按顺序出现.当 P 为星形多边

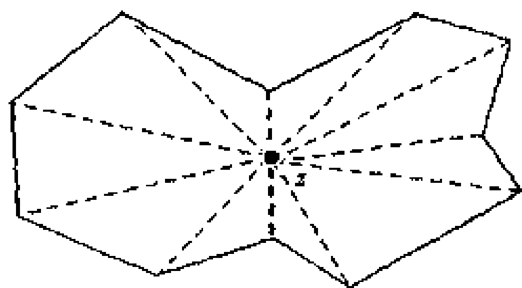


图 1-2

形时,如图1-2所示.关键是要找到点 z,使得 $\overline{zp_i}(i=1,2,\cdots,n)$ 整个在 P 内,这样的 z 点的集合称为 P 的核.可以证明,在线性时间内可以确定多边形 P 的核(若非空).这样,选择 z 点并产生楔形的时间限界不会超过线性时间.因此,用 $O(\mathrm{lb}n)$ 时间和 $O(n)$ 空间能回答 n 个顶点的星形多边形的包含问题,其预处理时间为 $O(n)$.

当 P 是任意简单多边形时,可以先将 P 分解为若干个凸多边形,然后再用上述方法回答多边形的包含问题.

问题 2　确定点 q 在平面剖分中的位置.

平面图总可以嵌入平面,它的边映射到直线段,这种嵌入的图称为平面直线图,记为 G.平面直线图 G 确定平面的一种剖分,若 G 的顶点度数不小于2,则该图的有界区域是简单多边形.假设 G 是连通的.

对平面剖分中的每个子区域逐一检查,可以确定点 q 所在的子区域,但这种方法耗费的时间多,不可取.为了加快查找定位点 q 的速度,一种有效的方法是将平面直线图 G 的所有有界区域组织成二叉树结构,然后进行对分查找.现有多种方法将 G 的有界区域组织成二叉树结构,下面仅以水平长条方法为例说明.

给定平面直线图 G,通过 G 的每个顶点画一条水平线,如果 G 有 n 个顶点并且没有两个顶点的 y 坐标相同,则平面被划分成 $n+1$ 个水平长条,见图1-3.依据顶点的 y 坐标分类这些水平长条,然后执行对分查找,耗费 $O(\mathrm{lb}n)$ 查找时间可以确定点 q 所在的水平长条.

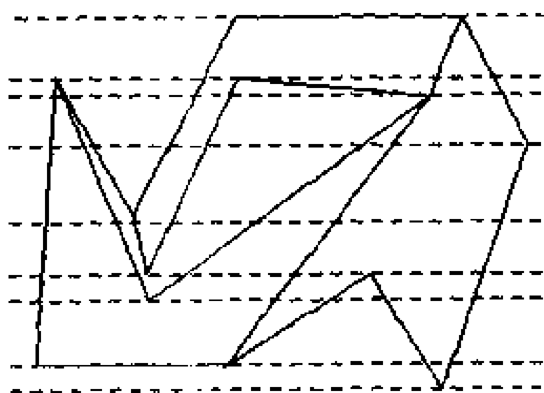


图 1-3

每个水平长条均被 G 的某些边划分成梯形或三角形,这些边可以从左至右排序,也就是说,位于同一水平长条中的梯形或三角形可以从左至右排序,然后利用对分查找,耗费 $O(\mathrm{lb}n)$ 查找时间可以确定点 q 所在的梯形.这种方法所需要的预处理时间是 $O(n^2)$,存储空间为 $O(n^2)$.

1.2　范围查找问题

范围查找问题是点定位问题的对偶问题.指定 d 维空间中一个域 D,称为询问域,范围询问是报告包含在 D 中点集 S 的子集 S',或者计数位于 D 内点集 S' 中的点数.前者使用集合并运算而后者使用加法运算.

d 维空间中的每一个维表示与该空间中点关联的一种信息,$d\geqslant 2$ 时,对域 D 的查找称为 d 重查找.比如查找某单位年龄在25岁至35岁之间,工资在1 000元至

2 000元之间的人员名单,便是2重查找.

询问域 D 是根据询问的性质来确定的,大多数询问域是矩形域、超矩形域(即不同坐标轴上区间的笛卡尔乘积,又称为正交询问)或者圆域. D 为圆域时,使用了邻近概念.为了回答范围询问,必须要建立 d 维空间中点集 S 的查找数据结构,这种数据结构分为静态的(一旦建立起便不再修改)和动态的(对项可以删除或插入)两种.建立这种数据结构将要耗费预处理时间和存储空间,此外,询问时间也是一种要考虑的时间开销.当静态的查找数据结构建立之后,主要考虑询问时间和存储空间.因此用耗费对(存储空间、询问时间)来描述范围查找方法的特征.对于实际问题的不同要求,可以选择相异的查找方法,使其在存储空间和询问时间之间进行折衷.

下面介绍两个范围查找算法.范围查找的最简单实例是一维范围查找问题:假设 n 个点分布在 x 轴上,询问范围是区间 $[a,b]$(称为 x 范围).利用对分查找方法(即平分被查找的有序集)可以有效地进行查找.这种方法采用均衡二叉树的数据结构.该方法的询问时间 $\theta(\text{lb}n+k)$ 和存储空间 $\theta(n)$ 都是最优的.

另外,考虑多维二叉树(k-D 树)的方法.

给定平面上 n 个点的点集 S,交替使用平行于 x 轴或 y 轴的直线将平面平分成两个矩形(有界的或者无界的),这些直线过 S 中的点,并且使直线两侧内 S 点的数目近似相等.这里的矩形是 x 区间 $[x_1,x_2]$ 和 y 区间 $[y_1,y_2]$ 的笛卡尔乘积 $[x_1,x_2]\times[y_1,y_2]$,其中 x_i、y_i($i=1,2$)可以为 $-\infty$, $+\infty$.显然,这是利用分治思想设计的一种方法.

上述划分平面的过程可以与一棵二维二叉树 T 联系起来.开始时,定义 T 的根并设置 R(根)为整个平面, S(根) $=S$;然后确定点 $p\in S$ 使 $x(p)$ 是 S(根)的点的 x 轴坐标的中值.过点 p 的直线(平行于 y 轴)把 S 分为两个大小近似相等的集合(平面被平分为两个半平面,用 R_1 和 R_2 表示), R_1 和 R_2 被赋给根的两个子结点.继续分割下去,当该过程达到一个不包含任何点的矩形时,终止该分割过程,对应的

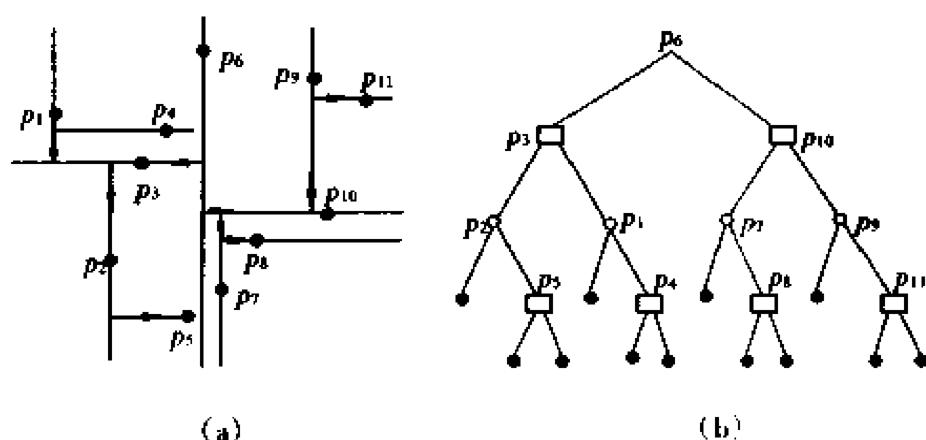


图 1-4

结点是 T 的叶结点.上述分割过程及对应的二维二叉树 T(即 2-D 树)如图 1-4 所示.

在图1-4中，$n = 11$，二叉树中有三种不同类型的结点：圆形结点表示垂直切割线的非叶结点；方形结点表示水平切割线的非叶结点；而实心结点表示叶结点.

范围查找中2-D树的用法如下：当关联于T的结点v的矩形域$R(v)$和矩形范围D的交非空时，过$p(v)$的直线$l(v)$把$R(v)$分割为两个矩形R_1和R_2.若$D \cap R(v)$整个包含在$R_i(i = 1,2)$中，则继续查找一个(区域，范围)对(R_i, D).另一方面，若$l(v)$分割$D \cap R(v)$，这表明$l(v)$和D的交非空，所以D可包含$p(v)$.因此，首先要测试$p(v)$是否在D的内部，如果在D的内部，则把它放入检索集合，然后继续查找两个(区域，范围)对(R_1, D)和(R_2, D).若查找工作达到叶结点，则终止范围查找.T的结点v带有三个参数$(p(v), t(v), M(v))$，其中点$p(v)$已被定义，另外两个参数共同确定直线$l(v)$，即$t(v)$表明$l(v)$是水平的或者垂直的，而且在第一种情况下，$l(v)$是直线$y = M(v)$，而在第二种情况下，$l(v)$是直线$x = M(v)$.算法把检索的点放在表U中(开始时U为空).$D = [x_1, x_2] \times [y_1, y_2]$表示询问范围.查找算法描述如下：

```
procedure SEARCH(v,D)
begin
  if  t(v) = 垂直的   then  [l,r] ← [x1,x2]
  else  [l,r] ← [y1,y2];
    if  l ≤ M(v) ≤ r  then
      if  p(v) ∈ D  then  U⇐p(v);
    if  v ≠ 叶结点   then
      if  l < M(v)  then  SEARCH(L son[v],D);
      if  M(v) < r  then  SEARCH(R son[v],D).
end
```

对图1-4所示例子的查找过程如图1-5所示，其图(a)中虚线表示询问范围D，

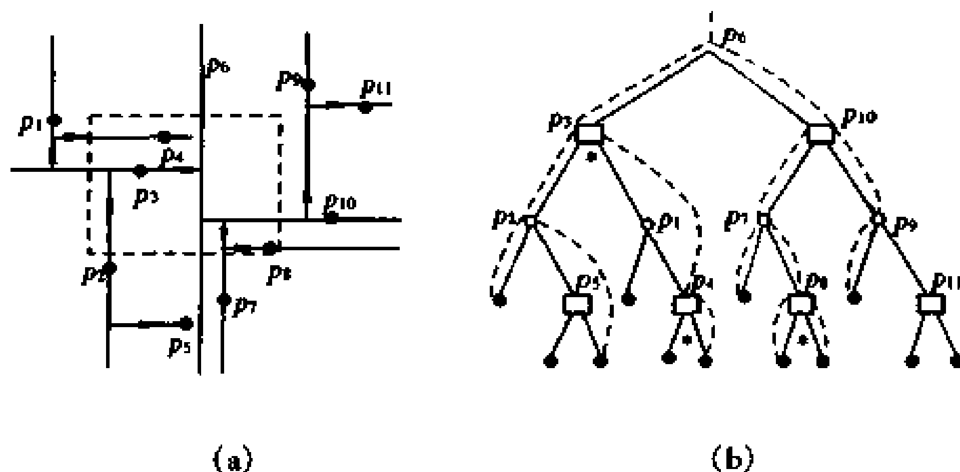


图1-5

图1-5(b)中虚线表示查找算法执行的T的结点的访问.在查找访问分叉的结点时(比如$p_6, p_3, p_2, p_4, p_{10}, p_7, p_8$)，要进行点是否包含在$D$内的测试.星号 * 标记的

结点是测试成功的结点，且取出该点.图中的检索集合为$\{p_3,p_4,p_8\}$.

该方法的预处理时间为$O(n\mathrm{lb}n)$，询问时间是$O(n^{\frac{1}{2}})$、存储耗费为$\theta(n)$.

1.3 判定点集是否在多边形内

本节将询问域D扩展为简单多边形P，判定点集S中哪些点落入多边形P内.下面介绍周培德提出的一个算法，该算法不需判断S中的每个点是否在多边形P的内部，其基本思想是：反复求剩余点集的凸壳C，只要判断凸壳C的顶点是否在P内及P的顶点(m个点)是否在C外，便可确定S中的哪些点位于多边形P的内部.

判定点集是否在多边形内部的算法：

输入：n个点$(q_1,q_2,\cdots,q_n)$的点集S.任意多边形P，其顶点序列为$p_1,p_2,\cdots,p_m$.

输出：点$q_1,q_2,\cdots,q_k(k\leqslant n)$位于$P$内.

步1 $i\leftarrow 1,S_i\leftarrow S,C'_i\leftarrow\varnothing,C''_i\leftarrow\varnothing,C'_0\leftarrow\varnothing,C''_0\leftarrow\varnothing$.

步2 求点集S_i的凸壳C_i，设$C_i=\{q_{i1},q_{i2},\cdots,q_{ii_1}\}$.

步3 if P的所有顶点在C_i的外部$\wedge$ C_i的所有顶点在P内

then 输出“$C'_1,C'_2,\cdots,C'_{i-1},S_i$在$P$内”，终止

else if P的所有顶点在C_i的外部$\wedge$ C_i的所有顶点在P的外部$\wedge$ P的边与C_i的边不相交

then 输出“$C''_1,C''_2,\cdots,C''_{i-1},S_i$在$P$外”，终止

else goto 步4

步4 设$C'_i=\{q_{i1},q_{i2},\cdots,q_{ij}\}$在$P$内，$0\leqslant j\leqslant i_1$，$j=0$时，$C'_i=\varnothing$.保留$C'_i$，$C''_i=C_i-C'_i$.

步5 $S_{i+1}\leftarrow S_i-C_i$，$i\leftarrow i+1$，重复步2,3,4步直到$S_i$为空集.

步6 输出“$C'_1,C'_2,\cdots,C'_{i-1}$在$P$内”，终止.

该算法在最坏情况下的复杂性为$\max[O(m\,n),O(ln\mathrm{lb}n)]$次比较和$O(ln)$次乘法，其中$l$是点集$S$的凸壳层数.

2 多 边 形

多边形分为凸多边形与任意多边形，这里所说的多边形是指简单多边形.本章将首先介绍凸多边形，然后介绍任意多边形(简称多边形)以及多边形的三角剖分和凸划分(划分成凸多边形).

2.1 凸多边形

凸多边形是一种特殊的多边形，它具有许多特性，比如凸多边形的所有顶点均

在任一条边的同一侧等,因此在许多问题中常常将多边形分割成凸多边形,特别是分割成三角形.本节介绍与凸多边形有关的问题的结果,包括判定点是否在凸多边形的内部,确定线与凸多边形的交,判定两个凸多边形是否相交以及确定两个凸多边形的交等.

定义1　由平面上若干条线段$\overline{p_1p_2},\overline{p_2p_3},\cdots,\overline{p_np_1}$围成的封闭有界域称为多边形.其中线段$\overline{p_ip_{i+1}}(i=1,2,\cdots,n,p_{n+1}=p_1)$称为多边形的边,相邻的两条边仅在端点相交,其交点$p_i$称为多边形的顶点,$\partial P$表示多边形$P$的边界.

定义2　多边形中具有共同端点的边称为相邻的边,比如$\overline{p_{i-1}p_i}$与$\overline{p_ip_{i+1}}$是两条相邻的边.若多边形中不相邻的边不相交,则称该多边形为简单多边形.

定义3　与同一顶点p_i关联的两条边$\overline{p_{i-1}p_i},\overline{p_ip_{i+1}}$形成的位于多边形内部的角$\angle p_{i-1}p_ip_{i+1}$称为多边形的内角.

定义4　若多边形P的顶点序列$p_1,p_2,\cdots,p_n$按逆时针方向排列,并且点p_{i+1}在$\overrightarrow{p_{i-1}p_i}$的左(右)侧,则称点$p_i$为凸(凹)点.以凸(凹)点为顶点的内角小于或等于$\pi(>\pi)$.

先求出凸多边形各顶点y坐标最小值所对应的顶点,设为p_i,p_i必为凸点.如果p_{i+1}在$\overrightarrow{p_{i-1}p_i}$的左(右)侧,那么该凸多边形顶点序列按逆(顺)时针方向排列.对任意简单多边形,将上述方法稍作修改,便可确定简单多边形顶点序列是逆时针排列还是顺时针排列。

凸多边形或多边形用其顶点序列$p_1,p_2,\cdots,p_n$(逆时针方向排列)来表示.另外,还可用均衡分层树的结构表示凸多边形,在这种表示下可得到一些重要结果.

定义5　如果

1° P_0是至多有4个顶点的多边形.

2° $P_k=P$.

3° 删去P_i的某些顶点可以得到P_{i-1}.

则多边形序列$P_0,P_1,\cdots,P_k$是凸多边形P的均衡分层表示.

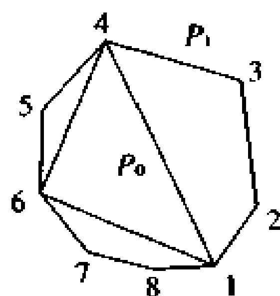


图2-1

图2-1是有8个顶点的凸多边形的均衡分层表示,多边形P_1由8个顶点组成,从P_1中删去顶点2,3,5,7,8得到多边形P_0,即P_0由顶点1,4,6组成.

如果把凸多边形边的序列存储在均衡树的叶中,即一个叶存储一条边,那么树的每一级对应分层表示中的一个多边形,如图2-2所示.图中根结点是多边形$P_0=\{1,5\}$,深度为1的结点表示多边形$P_1=\{1,3,5,7,1\}$,即根结点的左子结点表示用边链(1,3)、(3,5)代替P_0的边(1,5),右子结点表示用边链(5,7)、(7,1)代替P_0的边(1,5).依次类推.均衡树中所有内结点(叶除外)所表示的多边形恰好是凸多边形P的一种划分.

利用上述凸多边形的均衡分层表示可以证明下述结论.

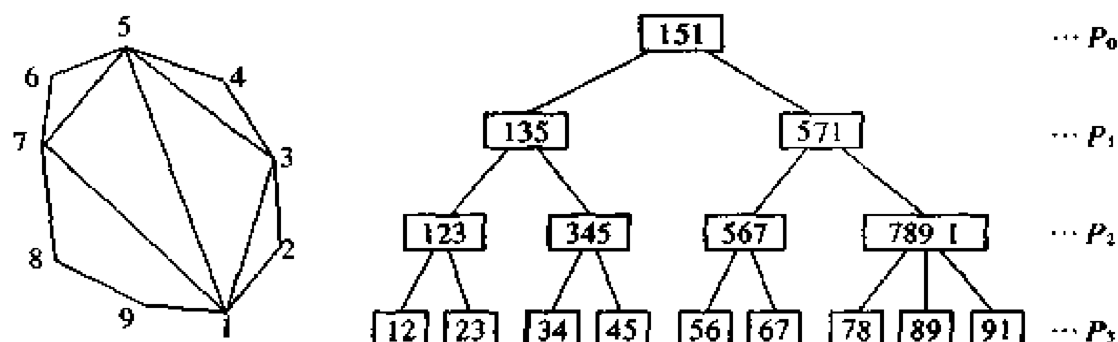


图 2-2

引理 1 (1) 在时间 $O(n)$ 内可以构造凸多边形的均衡分层表示,其中 n 是凸多边形 P 的顶点数;

(2) 如果 $P_0, P_1, \cdots, P_k$ 是 P 的均衡分层表示,那么 $k = O(\mathrm{lb}n)$.

定理 1 给定凸多边形 P 的均衡分层表示及 P 的内点 q 和任意点 x,在时间 $O(\mathrm{lb}n)$ 内可以确定 x 是否在 P 内,其中 n 是 P 的顶点数.

定理 2 给定凸 n 边形 P 的均衡分层表示和直线 L,在 $O(\mathrm{lb}n)$ 时间内可以确定 L 与 P 是否相交并确定 L 与 P 的交.

定理 3 给定凸 n 边形 P 的均衡分层表示和①线段 s,则在时间 $O(\mathrm{lb}n)$ 内可以计算 $P \cap s$;②点 p,则在时间 $O(\mathrm{lb}n)$ 内可以判定 p 是否在 P 内.

定理 4 给定凸 n 边形 P 和凸 m 边形 Q 的均衡分层表示,则在 $O(\mathrm{lb}(n+m))$ 时间内可以判定 P 与 Q 是否相交.

定理 5 设 P 与 Q 是凸 n 边形,则在 $O(n)$ 时间内可以计算 $P \cap Q$.

定理 1 至定理 5 均可用构造方法证明.

2.2 多 边 形

多边形是 S 计算几何研究的对象之一,本节介绍与多边形有关的某些问题.

问题 1 设某建筑物的平面图如图2-3所示(12个顶点的多边形),问至少要安装多少只监视器才能监视整座建筑物.

监视器不能穿透墙壁,每只监视器是从不同方向可以看见的一个固定点,它可以旋转角度 2π 监视其周围环境.图 2-3 中 4 个圆点表示 4 只监视器的位置.

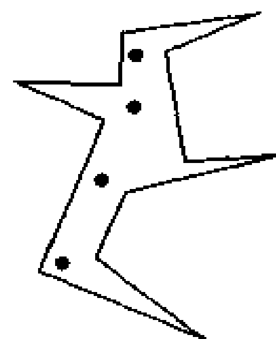


图 2-3

设点 x、y 位于多边形 P 内,从点 x 可以看见点 y(或者从 y 可以看见 x) 当且仅当封闭线段 $\overline{xy}$ 完全位于多边形 P 的内部:$\overline{xy} \subset P$. 线段 $\overline{xy}$ 称为可视线.

一只监视器是一个点,如果多边形 P 中每个点与一组监视器中至少一只监视器所在位置的点的连线是可视线,那么称该组监视器覆盖

多边形 P. 可以证明,覆盖12个顶点的任意多边形需要监视器的最大数恰好是4. 设 $f(P_n)$ 是覆盖有 n 个顶点的任意多边形 P_n 所需监视器的最小数

$$f(P_n) = \min_S |\{S \mid S覆盖P\}|,$$

其中 S 是点集, $|S|$ 表示 S 中元素的个数. 另设 $F(n)$ 表示所有 n 个顶点多边形 P_n 上函数 $f(P_n)$ 的最大值

$$F(n) = \max_{P_n} f(P_n).$$

定理6　$F(n) = \lceil n/3 \rceil$.

问题2　确定任意多边形的核.

下面是周培德提出的一种算法,该算法只扫描并处理 P 的凹点,对于 P 的凸点则不作什么处理,因而节省了计算时间.

确定多边形核的算法(部分描述):

输入:多边形 P 的顶点序列 $p_1, p_2, \cdots, p_n$(按逆时针方向排列).

输出:多边形 P 的核 $K(P)$ 的顶点序列.

步1　确定多边形 P 的凸凹顶点.

步2　if　$p_i(i = 1,2,\cdots,n)$ 是凸的　then　$K(P) = P$

　　else　求出 P 的凹点,并重新编号凹点: $q_1, q_2, \cdots, q_m$.

步3　$i \leftarrow 1, P_1 = P$.

步4　处理凹点 q_i:求与 q_i 关联的两条边和 P_i 边的交点,确定 P_{i+1}(q_i、交点、交点之间 P_i 的顶点、交点所围成的多边形)

步5　$i \leftarrow i+1$,重复步4,直至 $i = m+1$. 则多边形 P_m 即所求的核.

该算法的耗费为 $O(l\,n)$ 次乘法,其中 l 是多边形凹点的数目.

2.3　多边形的三角剖分

可以证明任意简单多边形都存在一个三角剖分,本节介绍构造多边形的三角剖分的算法.

定义6　设多边形 P 有 n 个顶点 $p_1, p_2, \cdots, p_n$, $\overline{p_ip_j}$ 是 P 的对角线,并且不与 P 的顶点及边相交,它划分 P 为两部分. 对 P 逐步增加不相交的对角线,直至 P 的内部全部被划分成三角形,这样的划分称为多边形的三角剖分.

定义7　设 p_{i-1}、p_i、p_{i+1} 是多边形 P 的三个连续顶点,并且 $\overline{p_{i-1}p_{i+1}}$ 是 P 的一条对角线,那么 p_{i-1}、p_i、p_{i+1} 构成 P 的一个耳, p_i 称为耳尖.

利用不断切去一个耳的方法可以将凸多边形三角剖分. 下面介绍周培德提出的任意简单多边形三角剖分的一种算法,该算法的基本想法是先将任意简单多边形划分成若干个凸多边形,然后对每个凸多边形进行三角剖分.

凸多边形三角剖分的算法 CPTA:

输入:凸多边形 P_1 的顶点序列 $p_1, p_2, \cdots, p_n$.

输出: P_1 的三角剖分序列 $p_1p_2p_i, p_2p_ip_j, \cdots, p_lp_{n-1}p_n$.

步1　计算 P_1 的直径,设直径的两个端点为 p_i 与 p_j.

步 2 比较 $\overline{p_{i-2}p_i}, \overline{p_{i-1}p_{i+1}}, \overline{p_ip_{i+2}}$ 的长度，取较短者作为对角线，删去相应的顶点(耳尖)，并输出相应的三角形(耳). 对 p_j 同样处理.

步 3 $P'_1 \leftarrow P_1 - (p_{i-1} \cup p_i \cup p_{i+1}) \cap (p_{j-1} \cup p_j \cup p_{j+1})$

步 4 对 P'_1 重复步 1 至步 4，直至 P_1 被分割完毕.

该算法的耗费为 $O(n^2)$，可以达到 $O(n)$.

任意多边形三角剖分的算法 APTA：

步 1 分割多边形 P 成凸多边形序列 $P_1, P_2, \cdots, P_k$.

步 2 对 $P_i(i = 1,2,\cdots,k)$ 执行 CPTA 算法.

该算法的复杂性可以达到 $O(n)$.

2.4 多边形的凸划分

本节介绍简单多边形划分成凸多边形(简称凸划分)的算法. 划分多边形成凸多边形有两个目标：① 划分多边形成尽可能少的凸多边形，② 尽可能快地完成划分工作，即设计时间复杂性低的算法.

用来划分多边形的线段有两种：多边形的对角线；端点在 ∂P 上并位于 P 内部的线段. 线段划分较对角线划分更复杂些. 凸多边形的数目与凹点的数目有下列关系.

定理 7 设 M 是划分多边形成凸多边形的最少数目，对于有 r 个凹点的多边形，则有 $\lceil \frac{r}{2} \rceil + 1 \leqslant M \leqslant r + 1$.

寻找最优凸多边形数目的划分比寻找次最优费时得多. 1983 年，格林(Greene)设计出最优凸划分的一种算法，时间复杂性为 $O(n^4)$. 1985 年，Keil 改进到 $O(n^3 \mathrm{lb} n)$. 周培德于 1997 年提出了一种划分多边形成凸多边形的算法，其时间复杂性是 $O(n)$.

定义 8 如果 p_i 是多边形 P 的凹点，$\overline{p_{i-1}p_i}$、$\overline{p_ip_{i+1}}$ 是与 p_i 关联的两条边，$\overline{p_{i-1}p_i}$($\overline{p_{i+1}p_i}$) 的延长线与 $\overline{p_ip_{i+1}}$($\overline{p_ip_{i-1}}$) 所夹的角域称为 p_i 的 $A(C)$ 域；$\overline{p_{i-1}p_i}$ 的延长线与 $\overline{p_{i+1}p_i}$ 的延长线所夹的角域称为 p_i 的 B 域. 图 2-4 中已示出凹点 p_2 的 A、B、C 域，顶点 p_4、p_8 分别落入 A、C 域，顶点 p_5、p_6、p_7 落入 B 域. 利用三阶行列式的值可以判断点落入哪个域.

周培德提出的算法思想如下：首先找出多边形 P 的凸、凹顶点，然后逐个处理凹点，即把凹点变成凸点，从而达到分割多边形为凸多边形的目的. 处理凹点 q_i 的方法是：划分 q_i 的周围域为 A、B、C 域，依据多边形其它顶点落入不同域将作不同处理，例如，凹点 q_j 落入 q_i 的 B 域并且 q_i 落入 q_j 的 B 域，$\overline{q_iq_j}$ 不与多边形边相交，那么 $\overline{q_iq_j}$ 分割多边形成两个多边形 P_1 与 P_2，并且在 P_1、P_2 内 q_i、q_j 不再是凹点. 只

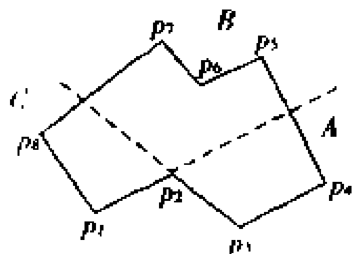


图 2-4

要连线位于某凹点的 B 域内，那么该凹点便丧失凹性而变成凸点．如果多边形凹点 q_i 的 B 域中无多边形顶点并且 P 的边 $\overline{p'p''}$ 的两端点分别位于 q_i 的 A、C 域中，则作两条连线，多边形 P 被分割成一个三角形和两个多边形 P_1 与 P_2，并且在 P_1 与 P_2 内 q_i 不再是凹点，直至消去所有凹点才终止．

3 凸 壳

凸壳是S计算几何中最普遍、最基本的一种构形，不仅它自身具有许多特性，而且它还是构造其它几何形体的有效工具．在应用中，许多实际问题可以归结为凸壳问题．本章介绍凸壳的基本概念及计算凸壳的算法．

3.1 凸壳的基本概念

设 S 是平面(E^2)中的点集，用CH(S)表示点集 S 的凸壳，而BCH(S)表示 S 的凸壳边界．

定义1 设 S 是平面上的非空点集，p_1、p_2 是 S 中任意两点，如果点

$$p = tp_1 + (1-t)p_2$$

属于 S，其中 $0 \leqslant t \leqslant 1$，则称 S 是凸集．

这就是说，如果 S 中任意两点所连线段全部位于 S 之中，那么 S 是凸的．

定义2 平面点集 S 的**凸壳** CH(S) 是包含 S 的最小凸集．

定义3 平面点集 S 的**凸壳边界** BCH(S) 是一凸多边形，其顶点为 S 中的点．

BCH(S) 是包围 S 的最小凸多边形 P，即不存在多边形 P'，使得 $P \supset P' \supset S$ 成立．

利用凸壳的概念可以简化许多问题的求解．例如，点 p 到点集 S 的距离可以用点 p 到 CH(S) 的距离来代替；点集 S_1 与 S_2 的距离可用 CH(S_1) 与 CH(S_2) 的距离代替；S 中最远两点的距离即 S 的直径以 CH(S) 的直径代替；点集 S_1 与 S_2 是否相交，可由 CH(S_1) 与 CH(S_2) 是否相交来决定．

$BCH(S)$ 是一凸多边形，其顶点数目 $m(k)$ 和点集中点的数目 n 有下列关系：

(1) 当 $n \to \infty$ 时，k 维球体中均匀独立地随机分布 n 个点，其凸壳顶点数

$$m(k) = O(n^{(k-1)/(k+1)}).$$

当 $k = 2$(平面情况)时，$m(2) = O(n^{\frac{1}{3}})$．

当 $k = 3$(空间情况)时，$m(3) = O(n^{\frac{1}{2}})$．

(2) 当 $n \to \infty$ 时，点集中的点呈 k 维正态分布，则点集凸壳的顶点数

$$m(k) = O((\text{lb}\,n)^{(k-1)/2}).$$

(3) 当 $n \to \infty$ 时，若 k 维空间中 n 个点的分量是独立地从任何连续分布的集合中随机选取，则其凸壳的顶点数

$$m(k) = O((\text{lb}\,n)^{k-1}).$$

总之，有 $\lim_{n\to\infty}\frac{m(k)}{n}=0$，这表明凸壳顶点数大大少于点集中的点数. 如用凸壳代替点集，那么不仅使得问题变得简单，而且可以节省存储空间.

计算平面点集 S 的凸壳，就是按逆时针方向计算边界顶点. 当 S 是平面上任意 n 个点的点集时，计算 CH(S) 至少耗费 $O(n\mathrm{lb}n)$ 时间. 如果 S 是平面上任意多边形顶点序列，那么耗费线性时间便可计算 CH(S). 当 S 中的点实时增加或减少时，耗费 $O(\mathrm{lb}^2n)$ 时间可以求得 CH(S)(即保持凸壳).

3.2　计算凸壳的算法(二维)

本节介绍寻求平面点集凸壳的算法，包括卷包果法、格雷厄姆(Graham) 方法、分治算法等.

卷包果法是由 Chand 和 Kapur 于 1970 年提出的，其基本思想如下：首先过 y 坐标最小的点 p_1 画一水平直线 l，显然该点是凸壳的一个顶点. 然后 l 绕 p_1 按逆时针方向旋转，碰到 S 中的第 2 个点 p_2 时，直线 l 改绕 p_2 按逆时针方向旋转而在 p_1 与 p_2 之间留下一线段，该线段为凸壳的一条边. 继续旋转下去，最后直线 l 旋转 360° 回到 p_1，便得到所要求的凸壳.

直线 l 绕点 p_i 的旋转是通过以下方法实现的：首先连接 p_i 与非凸壳顶点 p_j，$j=i+1,i+2,\cdots,n$，得到线段 $\overline{p_ip_j}$，然后求这些线段与 l(即 $\overline{p_{i-1}p_i}$) 的夹角，组成最小夹角的另一顶点 p_{i+1} 即凸壳顶点.

卷包果法的时间复杂性为 $O(n^2)$.

1972 年，格雷厄姆提出一种求平面点集凸壳的方法，称为**格雷厄姆方法**. 其步骤如下：

(1) 设点集中 y 坐标最小的点为 p_1，把 p_1 同点集中其它各点用线段连接，并计算这些线段与水平线的夹角. 然后按夹角大小及到 p_1 的距离进行词典式分类，得到一序列 $p_1,p_2,\cdots,p_n$，依次连接这些点，便得一多边形. p_1 点是凸壳边界的起点，p_2 与 p_n 也必是凸壳顶点. $p_{n+1}=p_1$. 图 3-1 为格雷厄姆方法的解释：图(a) 依各点倾角的大小分类. 图(b) 按顺序连线成一多边形. 图(c) $k=4$，向后倒查，p_1 与 p_4 在 $\overline{p_{k-1}p_{k-2}}$ 两侧，所以 p_3 不在边界上，删去 p_3. 图(d) 原 p_4 成为 p_3，p_1 和 p_4 在 $\overline{p_{k-1}p_{k-2}}$ 同侧，p_3 暂时在边界上，继续查下去，图(e) 最后得到凸壳的 6 个顶点.

(2) 删去 p_3、p_4、$\cdots$，p_{n-1} 中不是凸壳顶点的点，方法如下：

```
begin
    1. k ← 4
    2. j ← 2
    3. if   p1 和 pk 分别在 p(k-j+1)p(k-j) 两侧
            then 后继顶点编号减 1, k ← k - 1, j ← j - 1
       else  p(k-1) 暂为凸壳顶点，并记录
    4. j ← j + 1, goto  3, 直至 j = k - 1
    5. k ← k + 1, goto  2, 直至 k = n + 1
```

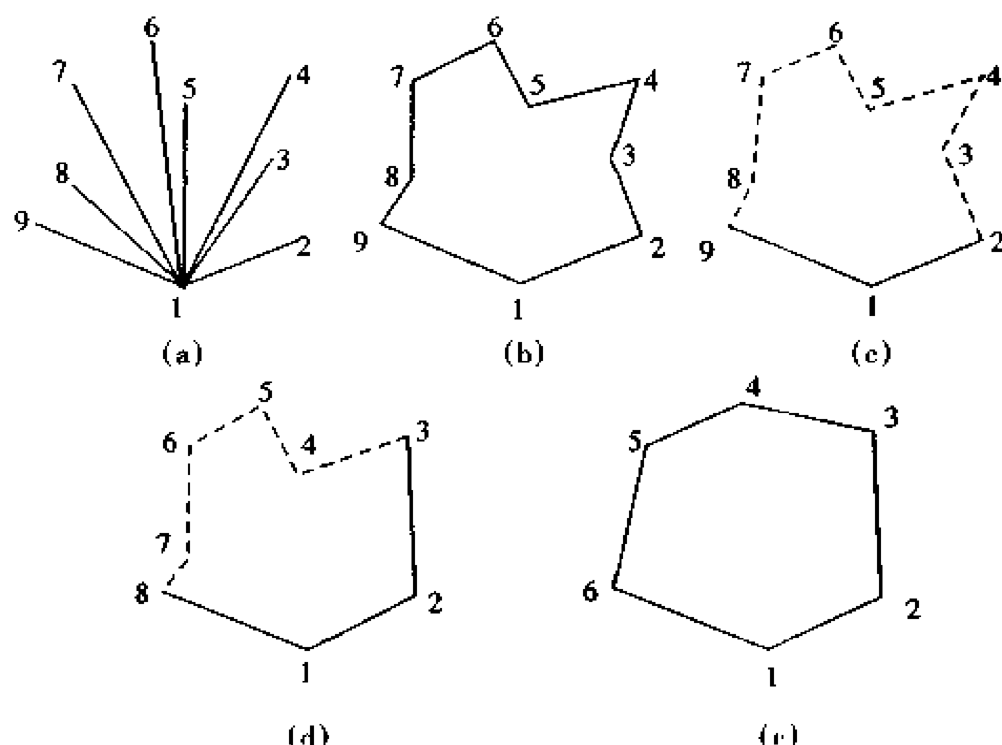


图 3-1

end

(3) 顺序输出凸壳顶点.

格雷厄姆方法的时间复杂性为 $O(n\mathrm{lb}n)$.

Preparata 和 Hong(1977 年) 把分治技术首先应用于凸壳问题.把点集 S 分成两个大小近似相等的子集 S_1 和 S_2,然后分别递归地寻求 CH(S_1) 和 CH(S_2),这是两个凸多边形,设为 P_1 和 P_2,最后找 $P_1 \cup P_2$ 的凸壳.这便是分治算法求点集 S 的凸壳的基本思路,算法描述如下:

步 1　把 S 中的点按 x 坐标分类.

步 2　分 S 成两个子集 S_1 和 S_2,S_1 和 S_2 分别包含$\lceil \frac{n}{2} \rceil$和$\lfloor \frac{n}{2} \rfloor$个点.

步 3　分别计算 $P_1 = CH(S_1)$ 和 $P_2 = CH(S_2)$.

步 4　合并 $CH(S_1)$ 和 $CH(S_2)$,即计算 $CH(CH(S_1) \cup CH(S_2))$.

分治法求 n 个点的集合的凸壳所需要的时间为 $O(n\mathrm{lb}n)$.

3.3　凸壳的应用

本节介绍一些实际问题,这些问题从表面上看与凸壳没有什么关系,但只要作些技巧性的转换,就变成了求凸壳问题.本节所示算法均为周培德提出的.

问题 1　确定任意多边形的凸、凹顶点.

确定任意多边形各顶点的凸、凹性，将为分割任意多边形成凸多边形提供有利条件. 如果多边形顶点序列按逆时针方向排列，并且顶点 p_{i+1} 在 $\overline{p_{i-1}p_i}$ 的左（右）侧，那么顶点 p_i 是凸（凹）的，这种方法的时间复杂性为 $O(n)$.

下面利用分割求解的思想设计了一种算法，其基本想法是，先找出任意多边形各顶点组成的点集的凸壳，该凸壳的顶点必是原多边形的部分顶点，并且这些顶点是凸的. 设 p'_j 是凸壳顶点，$j=1,2,\cdots,m$. 这 m 个点将多边形顶点序列 $p_1,p_2,\cdots,p_n$ 分成 m 个子序列，比如：

$$p_1=p'_1\underbrace{,p_2,\cdots,}_{1}\quad p_i=p'_2\underbrace{,p_{i+1},\cdots}_{2},\cdots,\quad p_l=p'_m\underbrace{,\cdots,p_n}_{m}$$

然后确定这些子序列组成的点集（即点链）$S^k_{j_k,(j+1)_k}$，并对点集 $S^k_{j_k,(j+1)_k}\cup\{p_j,p_{j+1}\}$ 分别再运用求凸壳的算法，新增的凸壳顶点必是原多边形的凹点. 算法反复执行分割顶点序列与求凸壳的工作，便可确定原多边形剩余顶点的凸凹性，直至所有点集 $S^k_{j_k,(j+1)_k}$ 为空. 该算法在最坏情况下的时间复杂性为 $O(n^2)$.

问题 2　利用凸壳求解货郎担问题.

货郎担问题是指给定 n 个点及任意两点之间的距离，求经过每个点恰好一次而且总长度最小的回路. 货郎担问题是一个 NP 难问题，现已有许多近似方法求解. 这里介绍的算法是依据几何中的基本运算和算法完成的. 基本作法是先求出点集的凸壳，以点集中的点与凸壳各边界的距离的最小值为标准划分点集中的点为不同类，此时注意了位于相邻两类间边界上及其附近的点与点团的处理. 然后，对各类中的点集分别重复求凸壳与点分类的工作，直至所有子类为空. 此时便得到一条完整的路径. 该算法已用于求中国31个省会城市的回路问题，得到一条长 15 492km 的回路. 可以进一步改进上述方法，求得长度为 15 460 km 及 15 440km 的回路. 如果重复执行该算法并增加一些技巧，那么可以求得长度为 15 404km 的最短回路，而时间复杂性不超过 $O(n^3)$.

问题 3　凸多边形直径.

确定凸多边形直径的算法：

输入：凸多边形 L 的顶点序列 $p_1,p_2,\cdots,p_n$（逆时针方向排列）.

输出：凸多边形 L 的直径 $d=|\overline{p_ip_j}|$.

步 1　for　$i=1$　to　n　do

$$q_{ix}\leftarrow\frac{p_{(i+1)x}+p_{ix}}{2},\quad q_{iy}\leftarrow\frac{p_{(i+1)y}+p_{iy}}{2}.$$

步 2　$i\leftarrow 1$.

步 3　$j\leftarrow i+1$.

步 4　计算夹角 $\angle p_iq_ip_j$（$p_{n+1}=p_1,\cdots,p_{n+k}=p_k$；$q_{n+1}=q_1,\cdots,q_{n+k}=q_k$）.

步 5　if　$\angle p_iq_ip_j<\dfrac{\pi}{2}$　then　$j\leftarrow j+1$，转入步 4，

　　else　计算距离 $|\overline{p_ip_j}|$（记为 d_i）.

步 6　if　$i=n$　then　$d\leftarrow\max(d_1,d_2,\cdots,d_n)$，输出 d，终止.

else $i \leftarrow i+1$,转入 步3.

该算法的复杂性为 $n-1$ 次比较,n 次求距离运算和 $n^2/2$ 次求夹角运算.

4 沃罗诺图及其应用

沃罗诺(Voronoi)图是仅次于凸壳的一个重要的几何结构,这是由于沃罗诺图在求解点集或其它几何对象与距离有关的问题时起着重要作用.狄里克莱(Dirichlet)(1850年)和沃罗诺(1908年)在他们的论文中都讨论过沃罗诺图的概念.

设想在一大片林区内设置 n 个火情观察塔 $p_1,p_2,\cdots,p_n$,每个观察塔 $p_i(i=1,2,\cdots,n)$ 负责其附近林区 $V(p_i)$ 的火情发现及灭火的任务.$V(p_i)$ 由距 p_i 比距其它 $p_j(j=1,2,\cdots,n,j\neq i)$ 更近的树组成,$V(p_i)$ 就是关联于 p_i 的一个沃罗诺多边形,而沃罗诺图由所有 $V(p_i)$ 组成($i=1,2,\cdots,n$).

如果把上述 n 个观察塔换成 n 个火源,这 n 个火源同时点燃,并以相同的速度向所有方向蔓延,那么火势熄灭处所形成的图便是沃罗诺图.

还有许多问题都可以归结为沃罗诺图,或利用沃罗诺图求解.

本章介绍什么是沃罗诺图,沃罗诺图的对偶图,沃罗诺图的性质,最近和最远意义下的沃罗诺图,构造沃罗诺图的算法及沃罗诺图的应用.

4.1 沃罗诺图的基本概念

设 p_1、p_2 是平面上两点,L 是线段 $\overline{p_1p_2}$ 的垂直平分线,L 将平面分成两部分 L_l 和 L_R,位于 L_l 内的点 p_l 具有特性:$d(p_l,p_1)<d(p_l,p_2)$,其中 $d(p_l,p_1)$ 表示 p_l 与 p_1 之间的欧几里德距离.这意味着,位于 L_l 内的点比平面其它点更接近点 p_1,换句话说,L_l 内的点是较平面上其它点更接近于 p_1 的点的轨迹,记为 $V(p_1)$.如果用 $H(p_1,p_2)$ 表示半平面 L_l,而 $L_R=H(p_2,p_1)$,则有 $V(p_1)=H(p_1,p_2)$,$V(p_2)=H(p_2,p_1)$.

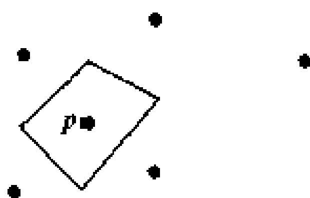


图 4-1

给定平面上 n 个点的点集 S,$S=\{p_1,p_2,\cdots,p_n\}$.定义 $V(p_i)=\bigcap_{i\neq j}H(p_i,p_j)$,即 $V(p_i)$ 表示比其它点更接近 p_i 的点的轨迹是 $n-1$ 个半平面的交,它是一个不多于 $n-1$ 条边的凸多边形域,称为关联于 p_i 的沃罗诺多边形或关联于 p_i 的沃罗诺域.图4-1中表示关联于 p_1 的沃罗诺多边形,它是一个四边形,而 $n=6$.

对于 S 中的每个点都可以作一个沃罗诺多边形,这样的 n 个沃罗诺多边形组成的图称为沃罗诺图,记为 Vor(S),如图4-2所示.该图中的顶点和边分别称为沃罗诺顶点和沃罗诺边.显然,$|S|=n$ 时,Vor(S) 划分平面成 n 个多边形域,每个多边形域 $V(p_i)$ 包含 S 中的一个点,而且只包含 S 中的一个点.Vor(S) 的边是 S 中某点对的垂直平分线上的一条线段或者半直线,从而

为该点对所在的两个多边形域所共有．Vor(S) 中有的多边形域是无界的．

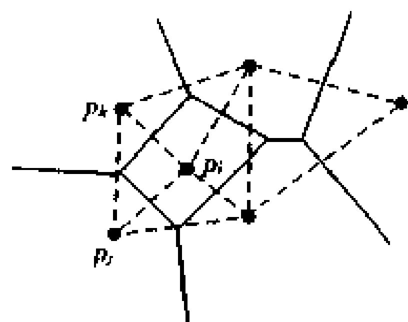


图 4-2

定理 1　$V(p_i)$ 是无界的，当且仅当 $p_i \in$ BCH(S)．

定理 2　n 个点的点集 S 的沃罗诺图至多有 $2n-5$ 个顶点和 $3n-6$ 条边．

定理 3　每个沃罗诺点恰好是三条沃罗诺边的交点．

定理 4　沃罗诺图的直线对偶图是 S 的一个三角剖分．

除上述定义的沃罗诺点、沃罗诺图之外，还有另外定义的沃罗诺点和沃罗诺图，以这些点为圆心，作过 S 中三个点的圆，该圆正好包含 S 其它全部点，这种沃罗诺点称为**最远点意义下的沃罗诺点**．这些最远点意义下的沃罗诺点及相应的无限凸多边形组成**最远点意义下的沃罗诺图**．

显然，只有点集 S 的凸壳边界上三点才可能组成圆把 S 中其它点包括进去．最远点意义下的沃罗诺图也有对偶图，该对偶图与凸壳也有共同的边界．

图4-3表示了10个点的集合 S 的两种沃罗诺图及其对偶图，其中图(a)为最近点意义下的沃罗诺图(虚线)及其对偶图(实线)．图(b)是最远点意义下的沃罗诺图(虚线)及其对偶图(实线)，该沃罗诺图具有5个沃罗诺顶点 A、B、C、D 与 E，7个无限凸多边形 ①、②、③、④、⑤、⑥ 与 ⑦．

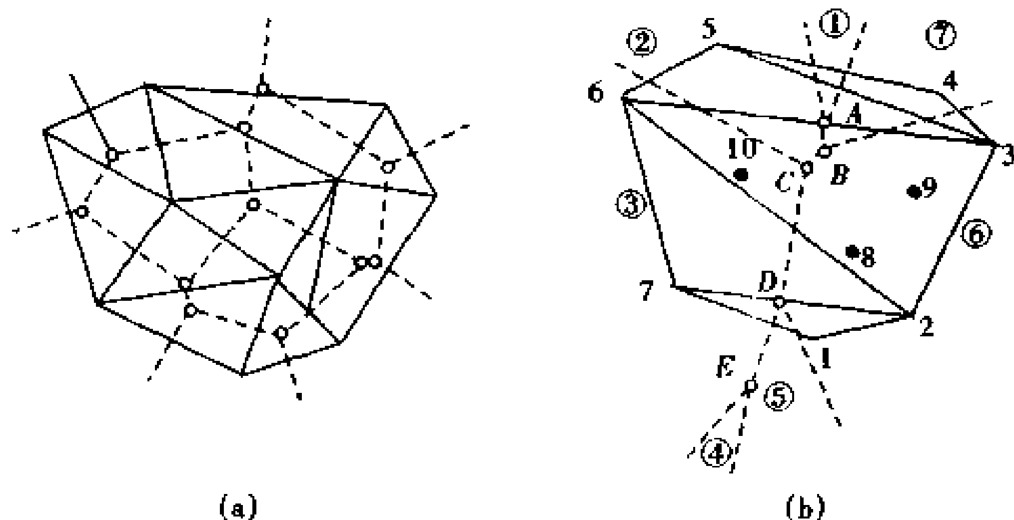


图 4-3

4.2　构造沃罗诺图的算法

本节介绍构造平面点集 S 的沃罗诺图的几种算法．

1. 半平面交算法

利用等式 $V(p_i)=\bigcap_{i\neq j}H(p_i,p_j)$ 构造 $n-1$ 个半平面的交,得到点 p_i 的沃罗诺多边形,然后逐点构造各点的沃罗诺多边形便得到 S 的沃罗诺图.

利用半平面的交求沃罗诺图的算法:

步1 按 x 坐标分类 S 中各点,设为 $p_1,p_2,\cdots,p_n$.

步2 $i\leftarrow 1$.

步3 利用 $V(p_i)=\bigcap_{i\neq j}H(p_i,p_j)$ 求点 p_i 的沃罗诺多边形.

步4 $i\leftarrow i+1$,重复步3,直至 $i>n$.

该算法的时间复杂性为 $O(n^2)$.

2. 分治法

构造沃罗诺图的分治算法是由 Shamos 和 Hoey(1975) 提出的,复杂性为 $O(n\mathrm{lb}n)$.算法的基本思想是按点的 x 坐标的中值(或先按点的 x 坐标分类,然后从中间分割)分割点集 S 为 S_1 与 S_2,使 $|S_1|=|S_2|=\frac{1}{2}|S|$.如果 S_1、S_2 含点数目大于4,则继续分割点集,直至子点集规模小于或等于4.对每个子点集利用本节1中的方法求沃罗诺图,然后不断合并相邻子点集的沃罗诺图,直至得到 $\mathrm{Vor}(S)$.

算法描述如下:

步1 划分 S 为规模近似相等的两个子集 S_1 和 S_2.

步2 递归地构造 $\mathrm{Vor}(S_1)$ 和 $\mathrm{Vor}(S_2)$.

步3 构造折线 B,分开 S_1 和 S_2,并使 $d(a,v)=d(b,v)$,其中 $a\in S_1,b\in S_2$,v 为折线上的点.

步4 删去位于 B 左侧的 $\mathrm{Vor}(S_2)$ 的所有边及位于 B 右侧的 $\mathrm{Vor}(S_1)$ 的所有边,得到集合 S 的 Voronoi 图 $\mathrm{Vor}(S)$.

3. 平面扫描算法

构造沃罗诺图的平面扫描算法是由 Fortune(1987) 提出的,其复杂性是 $O(n\mathrm{lb}n)$.

为了说明平面扫描算法,先介绍下面的概念.

设点 p 位于三维坐标系的 xy 平面上,顶点在 p 并且其侧面以45°倾斜的圆锥体垂直于 xy 平面.如果将第3个变元看成是时间,那么以 p 为顶点的圆锥体表示以单位速度在 p 周围扩张的圆,t 个单位时间之后,圆的半径为 t.

考虑以点 p_1 和 p_2 为顶点的两个圆锥体 $\mathrm{Con}(p_1)$ 和 $\mathrm{Con}(p_2)$,它们在三维空间中相交成一条曲线,该曲线位于垂直于 xy 平面的平面上,这个平面与 xy 平面的交是 $\overline{p_1p_2}$ 的垂直平分线.因此,虽然两个圆锥体的交线在三维空间中是一条曲线,但该曲线投影到 xy 平面上却是一条直线,而且是 $\mathrm{Vor}(\{p_1,p_2\})$.同样,以点 p_1、p_2 和 p_3 为顶点的三个锥体的交(3条曲线)在 xy 平面上的投影形成 $\mathrm{Vor}(\{p_1,p_2,p_3\})$.

构造沃罗诺图的平面扫描算法的基本思想是,通过与 xy 平面成45°倾斜的平面 π 扫描锥体,π 与 xy 平面的交线作为扫描线 L.假设 L 平行于 y 轴,并且它的 x 坐

标是 l. 当 L 的 $x=0$(点 p_1、p_2 的 x 坐标为0)时,扫描开始. 随后,扫描平面 π 和扫描线 L 向右平移,在平移过程中,平面 π 切割 $Con(p_1)$、$Con(p_2)$ 分别形成抛物线 $Par(p_1)$ 和 $Par(p_2)$,然后将其投影到 xy 平面,称为抛物线波面. 随着 π、L 的右移,两条抛物线 $Par(p_1)$、$Par(p_2)$ 的交的轨迹构成一条抛物线 $Par(Con(p_1)\cap Con(p_2))$,该抛物线在 xy 平面上的投影即 $\overline{p_1p_2}$ 的垂直平分线,它是 $Vor(\{p_1,p_2\})$. 在平面 π 和线 L 已扫描过的部分,点 p_1 和 p_2 及部分 $Par(Con(p_1)\cap Con(p_2))$ 均已形成,而未扫描部分,$Par(Con(p_1)\cap Con(p_2))$ 的剩余部分还未形成. 直至平面 π 离开 $Con(p_2)$ 时,扫描终止,$Vor(\{p_1,p_2\})$ 完全形成.

4.3　平面点集的三角剖分

平面上给定 n 个点 $p_1,p_2,\cdots,p_n$,所谓平面点集三角剖分是指用互不相交的直线段连接 p_i 与 p_j, $1\leqslant i,j\leqslant n$, $i\neq j$,并使凸壳内的每一个区域是一个三角形.

由于三角剖分是一个平面图,它有 n 个顶点,因此该图至多有 $3n-6$ 条边. 如果能给出这些边的一个表,那么就得到问题的一个解. 对三角剖分问题可以提出许多约束条件,例如,最小角最大化、边的总长度最小化(称为最小权三角剖分)等. 在许多应用中,最好是三角形尽可能为"等边的",或边的总长度最小.

无论是最近点意义下的沃罗诺图,还是最远点意义下的沃罗诺图都与点集的三角剖分有着密切的关系:最近点意义下沃罗诺图的对偶图就是点集的一种三角剖分,因而由点集的三角剖分可以计算沃罗诺图;最远点意义下沃罗诺图的对偶图是点集凸壳(凸多边形)的一种三角剖分,由点集凸壳的三角剖分可以求得沃罗诺图(只要该沃罗诺图存在). 现有多种三角剖分的算法,其中常用的有贪心算法和德劳赖(Delaunay)三角剖分算法. 最近点意义下的沃罗诺图的对偶图实际上是点集的一种三角剖分,该三角剖分就是德劳赖三角剖分. 因此,在构造点集的沃罗诺图之后,再作其对偶图,便得到德劳赖三角剖分. 也可以用类似于构造沃罗诺图的递归过程直接构造德劳赖三角剖分,而且时间复杂性为 $O(n\text{lb}n)$.

另外,周培德于近期提出了一种平面点集三角剖分算法,其主要步骤如下.

输入:平面上 n 个点的坐标 (x_i,y_i), $i=1,2,\cdots,n$,该点集记为 S.

输出:n 个点的三角剖分链表 $L=(p_1,p_2,p_i)\rightarrow(p_2,p_i,p_j)\rightarrow\cdots\rightarrow(p_{n-2},p_{n-1},p_n)$.

步 1　求点集 S 的凸壳顶点,设 C_1 为凸壳. 然后求 $\{S-C_1\}$ 的凸壳顶点,设 C_2 是 $\{S-C_1\}$ 的凸壳. 继续下去,直至求得 C_m, C_m 内不含 S 中的点、包含 S 中的 1 个点或 2 个点. 然后分别转步 2、步 3 和步 4.

步 2　处理 C_m 内不含 S 中点的情况. 先求 C_m 的直径,然后调用算法 CPTA 三角剖分 C_m. 转步 5.

步 3　处理 C_m 内含 1 个点 p 的情况. 将 C_m 各顶点与 p 连接,并按连线长度排序:$d_1^m,d_2^m,\cdots,d_{m_m}^m$,其中 $d_1^m(p_1^m,p)$ 最大,p_i^m 是 C_m 的顶点,$i=1,2,\cdots,m_m$. 然后用类似于步 2 的方法三角剖分 C_m. 转步 5.

步4　处理 C_m 内含两个点 p_1 和 p_2. 连接 p_1 与 p_2. C_m 中位于 $\overline{p_1p_2}$ 左、右侧的点分别构成两个凸壳,用步2方法三角剖分它们. 转步5.

步5　$i \leftarrow m$.

步6　分割 C_i 与 C_{i-1} 之间的环域.

步7　$i \leftarrow i-1$,转步6,直至 $i=1$. 转步9.

步8　设 $p_1p_2p_3$ 与 $p_3p_2p_4$ 是两个有一条公共边 $\overline{p_2p_3}$ 的三角形.

　　if　$|\overline{p_2p_3}| \leqslant |\overline{p_1p_4}| \vee \overline{p_2p_3}$ 与 $\overline{p_1p_4}$,交于两个三角形的外部

　　　　then　不改变原来的三角剖分

　　else　if　$|\overline{p_2p_3}| > |\overline{p_1p_4}| \wedge \overline{p_2p_3}$ 与 $\overline{p_1p_4}$ 交于两个三角形的内部

　　　　　then　连接 p_1 与 p_4,删去线段 $\overline{p_2p_3}$.

步9　对以 C_m 中的边(或已变动的边)为公共边的三角形对,用步8的方法检查是否需要改变原有的三角剖分. 然后,沿 $C_{m-1},\cdots,C_2$ 的各条边(或已变动的边)寻找两个有公共边的三角形对,并用步8的方法检查是否需要改变原来的三角剖分,直至所有凸壳的边检查完. 终止.

该算法的时间复杂性是 $O(n^4/m^3)$,其中 m 为点集凸壳的层数. $m=1$ 时,复杂性与算法 CPTA 的复杂性相同.

4.4　沃罗诺图的应用

沃罗诺图可用于求解最近邻近、最大化最小角三角剖分、最大空圆、最小生成树、货郎担问题、中轴问题等. 下面仅以最小生成树为例进行说明.

点集 S 的最小生成树(MST)是连接 S 中所有点的最小长度的树,即最小生成树的结点恰好是 S 中的点. 两结点之间的连线以欧几里德长度度量时,该树称为欧几里德最小生成树(EMST). MST 有许多应用,例如,许多局域网络利用树的形式生成基结点,MST 是最小化总路线长度的网络拓扑.

考虑计算平面点集的 MST 问题,该问题可以转化为计算平面完全图 G 的 MST 问题,解决后者的一种方法是基于贪心思想设计的,该方法是不断添加还没有选取的最短边至树中,同时保持树(非循环性)的特性. 这个算法称为库鲁斯卡(Kruskal)算法,描述如下:

步1　按长度将 G 的所有边: $e_1,e_2,\cdots,e_{n(n-1)/2}$ 分类.

步2　$T \leftarrow \varnothing$, $i \leftarrow 1$.

步3　while　T 中结点数 $< n$　do

　　　if　$T+e_i$ 是树(非循环)

　　　　then　$T \leftarrow T+e_i$

　　$i \leftarrow i+1$.

库鲁斯卡算法中,符号 $T+e_i$ 表示树 T 与边 e_i 的并,而步1决定该算法的时间复杂性为 $O(\|E\|\operatorname{lb}\|E\|)$,其中 $\|E\|$ 是图 G 的边数目.

由于图 G 有 $n(n-1)/2$ 条边,因此分类的复杂性是 $O(n^2\operatorname{lb}n)$,故求解 MST 需要 $O(n^2\operatorname{lb}n)$ 时间. 如果用德劳赖三角剖分边的集合代替完全图的边集合,即利用

德劳赖三角剖分边来构造 MST,那么库鲁斯卡算法的复杂性为 $O(n\mathrm{lb}n)$.

5　几何体的交

实际应用中常常碰到交问题,比如,消除隐藏线的问题是构造两个多边形的交,如果没有构造多边形交的最优算法,那么就不可能有效地消除隐藏线.再比如,模式识别、导线和元件布局、线性规划等领域中都会碰到判定几何体是否相交或确定它们的交的问题.为了有效地解决这些问题,必须设计出有效的交算法.

由于两个几何对象相交,仅当一个几何对象至少包含另一个几何对象的一点,所以交算法涉及到包含的测试.本章介绍线段交的算法、凸多边形交的算法以及半平面交的算法.

5.1　线段相交的算法

给定平面上 n 条线段,确定其中任意两条线段是否相交;如果相交,则求出所有的交点.平面多边形可以看成是平面上线段的集合,判定两个多边形是否相交的问题可以变换为两个线段集合中的线段是否相交的问题,而且这种变换在多项式时间内可以完成,因此有

$$\text{测试多边形相交问题} \propto \text{测试线段相交问题}.$$

另外,多边形是否为简单多边形的测试问题也可以多项式变换到测试线段相交问题.

假设平面上给定 n 条线段,利用对每两条线段检查是否相交的方法,可以确定 n 条线段中哪些线段相交.这种时间复杂性为 $O(n^2)$ 的算法是最简单的,但当 n 很大时,这种方法将耗费相当多的时间.因此需要寻找新的算法.

利用平移的 y 轴可以对线集 L 中所有的线段建立全序关系.该次序关系仅以三种方式变化:遇见线段 s 的左端点,将 s 加入该次序关系;遇见线段 s 的右端点,此时从次序关系中删去 s;达到两条线段 s_1 与 s_2 的交点 p,在次序关系中于 p 处 s_1 和 s_2 交换位置.

如果平面上线段 s_1 和 s_2 相交,当移动的 y 轴接近线段 s_1 与 s_2 交点的 x 坐标时,s_1 与 s_2 成为全序关系中相邻的两条线段.不必对每条线段检查是否与其它所有线段相交,只要检查在全序关系中相邻线段是否相交即可.Bentley 和 Ottmann 依据该思想提出利用平面扫描方法判定哪些线段相交,其步骤如下:

步 1　按 x,y 词典式将 $2n$ 个端点分类,并把它们放入优先队列 E. $A \leftarrow \varnothing$.

步 2　while　$E \neq \varnothing$　do　$p \leftarrow \mathrm{MIN}(E)$.

步 3　if　p 是左端点

　　then　$s \leftarrow p$ 是端点的线段;s 插入 T;s_1、s_2 分别是 T 中与 s 相邻的线段;如果 s_1 与 s 及 s_2 与 s 相交,则 A 记录相交的线对.

步 4　if　p 是右端点

then　$s \leftarrow p$ 是端点的线段；s_1、s_2 分别是 T 中与 s 相邻的线段；如果 s_1 与 s_2 交于 p 的右边，则 A 记录线对 (s_1, s_2)；从 T 中删去 s.

步5　if　p 是一个交点

then　$(s_1, s_2) \leftarrow$ 交点是 p 的线段；在 T 中交换 s_1 和 s_2，并按插入线段处理 s_1 与 s_2.

步6　while　$A \neq \varnothing$　do　$(s, s') \leftarrow A$；$x \leftarrow s$ 和 s' 交点的横坐标；

if　x 不是 E 的成员　then　输出 (s, s')；INSERT(x, E).

该算法仅在相邻线段中寻找交点，且所有相邻的线段至少被检查一次，所以算法不会遗漏交点. 算法的时间复杂性为 $O((2n+k)\mathrm{lb}n)$，其中 k 是交点数.

确定平面上 n 条线段所有交点这一问题的时间下界是 $\Omega(k + n\mathrm{lb}n)$，因此上述算法不是最优的. 1988 年 Chazelle 和 Edelsbrunner 给出了一个时间复杂性为 $\theta(k + n\mathrm{lb}n)$ 的算法，因而是解决该问题的最优算法.

由于判定两个多边形是否相交以及判定多边形是否是简单多边形等问题可以多项式变换到判定平面线段相交问题，而判定平面线段相交问题在时间 $O(n\mathrm{lb}n)$ 内可以求解，所以判定两个多边形是否相交和判定多边形是否为简单多边形问题可以在时间 $O(n\mathrm{lb}n)$ 内求解.

上述算法中，只是告诉判定哪两条线段相交，而没有说明如何判定两条线段相交，下面通过引入两个函数 same 与 intersect 计算这个问题.

设给定平面上两条线段 s_1 与 s_2，s_1 的两个端点为 p_1 和 p_2，如果 p_1、p_2 分别在 s_2 的两侧，同时 s_2 的两个端点分别在 s_1 的两侧，则 s_1 与 s_2 相交. 否则不相交.

用下述形式表示点和直线段：

type　point: = record　x, y: integer,

type　line: = record　p_1, p_2: point.

函数 same(s: line; p_1, p_2: point): boolean, 此函数当 p_1、p_2 位于 s 的同一侧时，返回值"true"，否则返回值"false"；函数 intersect(s_1, s_2: line): boolean, 当线段 s_1 和 s_2 相交时，返回值"true"，否则返回值"false". 引入这两个函数之后，不必求 s_1 与 s_2 的交点也能判断 s_1 与 s_2 是否相交.

```
function   same(s:line;p1,p2:point):boolean
var   dx,dy,dx1,dx2,dy1,dy2:real
begin
     dx ← s.p2.x - s.p1.x
     dy ← s.p2.y - s.p1.y
     dx1 ← p1.x - s.p1.x
     dy1 ← p1.y - s.p1.y
     dx2 ← p2.x - s.p2.x
     dy2 ← p2.y - s.p2.y
     same ← (dx * dy1 - dy * dx1) * (dx * dy2 - dy * dx2) > 0
end
```

直线 s 的方程表示为

$$\Delta(x,y) = dx \cdot (y - s.p_i.y) - dy \cdot (x - s.p_i.x) = 0, \quad i = 1,2.$$

凡在 s 上的点便使 $\Delta(x,y)=0$,而 s 两侧的点必有 $\Delta(x,y)\neq 0$(一侧使 $\Delta(x,y)>0$,另一侧使 $\Delta(x,y)<0$,所以如果 p_1、p_2 在 s 的同一侧,必然使 $\Delta(p_1.x,p_1.y)$ 和 $\Delta(p_2.x,p_2.y)$ 同时为正或负,即 p_1 与 p_2 在 s 的同侧的充分必要条件是

$$\Delta(p_1.x,p_1.y)\cdot\Delta(p_2.x,p_2.y) > 0.$$

同理,p_1 与 p_2 在 s 的异侧的充分必要条件是

$$\Delta(p_1.x,p_1.y)\cdot\Delta(p_2.x,p_2.y) < 0.$$

```
function  intersect(s1,s2:line):boolean
begin
    intersect ← not same(s1,s2.p1,s2.p2)
              and not same(s2,s1.p1,s1.p2)
end
```

如果两条线段的两个端点都在对方线段的两侧,则此两线段相交.

5.2 凸多边形的交

本节介绍凸多边形交的算法.平面上给定两个凸多边形 $P=\{p_1,p_2,\cdots,p_n\}$ 与 $Q=\{q_1,q_2,\cdots,q_m\}$,其顶点按逆时针方向排列,确定它们的交,记为 $P\cap Q$. $P\cap Q=\{q \mid q\in P \wedge q\in Q\}$.显然,两个凸多边形的交是一个凸多边形.如果 P、Q 分别有 n 和 m 个顶点,那么 $P\cap Q$ 至多有 $n+m$ 个顶点.下面介绍3种求 $P\cap Q$ 的算法.

确定凸多边形 $P\cap Q$ 的穷举算法:

输入:P 的边 $e_1,e_2,\cdots,e_n$ 及顶点序列 $p_1,p_2,\cdots,p_n$,Q 的边 $e_1',e_2',\cdots,e_n'$ 及顶点序列 $q_1,q_2,\cdots,q_m$.

输出:$P\cap Q$ 的边 $e''_1,e''_2,\cdots,e''_l$ 及顶点序列 $r_1,r_2,\cdots,r_l$.

```
for  i = 1  to  n  do
  for  j = 1  to  m  do
    begin
        if  ei 与 e'j 相交
          then  求交点并记录交点
    end
write  P∩Q 的边序列及顶点序列
```

该算法的时间复杂性为 $O(mn)$.

梯形交组成 $P\cap Q$ 的算法:

预处理:将 P、Q 顶点按 x 坐标分类.

步1　过 p_i、q_j 作垂直于 x 轴的直线($i=1,2,\cdots,n$,$j=1,2,\cdots,m$),形成数目不超过 $n+m-1$ 个长条域.

步2 对每个长条域，计算两个梯形的各顶点(有的长条域内只有一个梯形或一个三角形)及两个梯形的交，交记为 w_i.

步3 计算 $P \cap Q = \bigcup w_i$.

该算法的时间复杂性是 $O((m+n)\mathrm{lb}(m+n))$.

沿 P 边和 Q 边行进寻找 $P \cap Q$ 的算法：

```
begin
  i ← 1, j ← 1, k ← 1
  repeat
    begin
      if  p_i p_{i+1} 与 q_j q_{j+1} 相交   then   记录交点
       增加 i 或增加 j(沿 P 边或 Q 边行进)
      k ← k + 1
    end
  until   k = 2(n + m)
  if   没有交点   then
    begin
      if   p_i ∈ Q   then   P ⊆ Q
      else   if   q_i ∈ P   then   Q ⊆ P
         else   P ∩ Q = ∅
    end
end
```

图 5-1

沿 P 边或 Q 边行进的方法如下：开始时，选任一条边，比如选 Q 边行进，然后，当 Q 的现时边 $\overline{q_j q_{j+1}}$ 上已找到一个交点时，将改沿 P 边上行进，或当 P 的现时边 $\overline{p_i p_{i+1}}$ 上已找到一个交点时，将改沿 Q 边上行进，或者 P 的现时边上的所有交点已被找到，而 Q 的现时边仍有交点未找到，则在 P 边上行进. 见图 5-1 所示，图中 $\overline{p_{i+1} p_{i+2}}$ 上两个交点已找到，而 $\overline{q_{j+1} q_{j+2}}$ 上仍有交点未找到，则在 P 边上行进. repeat 语句循环 $2(n+m)$ 次，每次循环耗费常数时间，因此 repeat 循环耗费 $O(n+m)$ 时间. 其它语句的复杂性不超过线性时间，该算法的时间复杂性为 $O(n+m)$.

5.3 半平面的交及其应用

本节介绍求半平面交的算法，两个变量的线性规划问题及算法.

平面上给定 n 个半平面，它们由下述不等式确定

$$a_i x + b_i y + c_i \leqslant 0, i = 1, 2, \cdots, n,$$

目的是要求这 n 个半平面的交. 显然，所要求的交是一个凸多边形区域. 上述的 n

个不等式称为约束条件，满足 n 个不等式的解(即点)称为可实现解，并且它们的轨迹称为可实现区域(凸多边形区域)．图 5-2 中的阴影部分是线性约束下的可实现区域，虚线表示多余的约束．

假设已求得前 i 个半平面的交，它是一个至多 i 条边的凸多边形区域 A_i．A_i 与第 $i+1$ 个半平面的交是用线 $a_{i+1}x+b_{i+1}y+c_{i+1}=0$ 切割 A_i，并且保留右边部分或左边部分得到的，这个工作需要 $O(i)$ 时间．i 从 2 增至 n 时便可求得可实现域．其时间复杂性为 $\sum_{2}^{n} i = O(n^2)$．

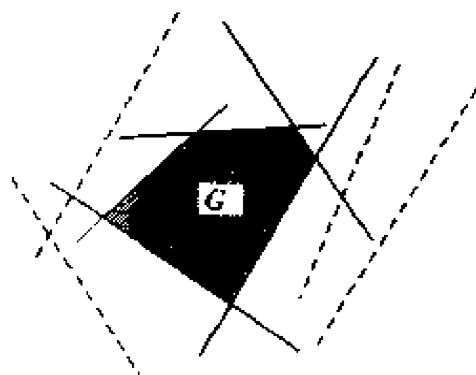


图 5-2

利用分治法也可以求解这个问题．设平面上给定 n 个半平面 H_i，$i=1,2,\cdots,n$，要求构造它们的交．

$$H_1 \cap H_2 \cap \cdots \cap H_n.$$

借助结合律，可以写成

$$(H_1 \cap \cdots \cap H_{n/2}) \cap (H_{n/2+1} \cap \cdots \cap H_n),$$

两个括号分别表示 $n/2$ 个半平面的交，它们至多是 $n/2$ 条边的凸多边形区域 A 与 A'．两个 k 条边的凸多边形区域能交 $O(k)$ 次，所以合并 A 与 A' 成一个凸多边形区域需要时间 $O(n)$．

求 n 个半平面 H_i 交的分治算法：

输入：由有向线段 s_i 确定的 n 个半平面 H_i，$i=1,2,\cdots,n$．

输出：凸多边形区域 $H = H_1 \cap H_2 \cap \cdots \cap H_n$．

步 1　将 n 个半平面 H_i 分成两个大小近似相等的集合．

步 2　递归地构造凸多边形区域 A 与 A'．

步 3　合并 A 与 A' 成 H．

令 $T(n)$ 表示用分治算法构造 n 个半平面的交所需要的时间，则有

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n).$$

解得

$$T(n) = O(n\operatorname{lb} n).$$

因此，用 $O(n\operatorname{lb} n)$ 时间可以构造 n 个半平面的交，而且这是最优的．

另外，简要介绍两个变量的线性规划．

二维线性规划问题：

目标函数　$f(x,y) = ax + by \rightarrow$ 最小(或最大)　(5-1)

约束条件　$\begin{cases} a_i x + b_i y + c_i \geqslant 0, & i = 1,2,\cdots,n; \\ x \geqslant 0, y \geqslant 0 \end{cases}$　(5-2)

线性规划问题是在取正值或 0 值的两个变量 (x,y) 的 n 个联立线性不等式的约束条件下，求(5-1)式的最小(或最大)值．线性规划问题的可实现区域是满足(5-2)式中约束的点 (x,y) 的集合，而且是 n 个半平面的交．可实现区域显然是一

个凸多边形区域,该凸多边形的顶点使目标函数减至最小或增至最大.

求解线性规划问题已有许多方法,这里提出的算法是依据分治思想设计的非数值算法,即首先将约束条件根据其斜率及在 x 和 y 轴上的截距进行分类,然后删去每一类中的多余约束条件,并求少量剩余约束条件对应的线链,最后求3条线链的交点,便可求得最优值.这种方法的优点在于不需要检验每个约束条件(多余的约束条件较早就被删去);求直线交点的次数达到最少,一般对各类线簇只需求少数几个交点;一般情况下,不必构造可实现区域多边形的所有边界;函数值的求值次数也达到最少.

6 矩形几何

由于矩形可以看成是特殊的凸多边形,所以第5章关于凸多边形交的算法也可以用于矩形的交.但是矩形具有其特殊的性质(它的边是相互垂直或平行的线段),因此应该设计出满足这种特殊结构的更有效的算法.

正交线段和矩形是构成许多应用的基础,比如,集成电路制造中所用的掩模常表示为边平行于坐标轴的一组矩形,任务是证实设计规则,要求均能被满足.对此任务稍加变换就可以成为一个矩形几何问题(例如矩形交问题).数据库中由并发控制产生的死锁问题的解决也导致出一个矩形并问题,等等.

本章着重介绍解决几个矩形几何问题的算法,包括矩形并的面积的算法、矩形并周长的算法、以及矩形交的算法.

6.1 矩形几何问题的特征及解决问题的途径

假设平面上给定 n 个矩形 $R_i(i=1,2,\cdots,n)$,R_i 的边分别平行于 x 轴或 y 轴,这样的 n 个矩形称为同等安置的矩形,它们的边可以看成是正交的平行线段的集合.同等安置矩形的特征是把平面划分成条状域,在每个条中只有一个变量,即是一维的.

矩形并的面积和并的周长问题具有一个共同的性质:给定问题的解可以和两个半平面(一条垂直线划分平面为两个半平面)的每个半平面中相似子问题的解联系起来.比如,矩形交问题的解是子问题解的并,矩形并的面积、周长等问题的解是子问题解的算术和.在这些情况中,平分线(扫描线)和矩形集合的交(扫描线上的一线段)包含了部分解的信息,并且扫描线左半平面得到的解不可修改,它是最后解的一部分,随着扫描线右移,扫描线左侧解得到推广,直至获得整体解.这既是解决矩形几何问题的一条途径,也是平面扫描方法的特性.由于矩形所具有的特征,这里的平面扫描不用优先队列而用线段树.

用区间[$B(v)$,$E(v)$](B 为左端点,E 为右端点)和其它参数来描述线段树的每个结点 v 的特征,其中结点计数 $c(v)$ 表示当前分配给 v 的线段的数目.参数 $c(v)$ 对所有矩形问题都是需要的,但针对不同矩形问题要增加不同的参数.

6.2 矩形并的面积与周长

给定平面上 n 个同等安置的矩形 $R_1, R_2, \cdots, R_n$，定义 n 个矩形的并 $F = R_1 \cup R_2 \cup \cdots \cup R_n$ 为

$$F = \{p \mid p \in R_1 \text{或} p \in R_2 \text{或} \cdots \text{或} p \in R_n\}.$$

定义 F 的周长为边界长，F 的面积是边界所围域的面积. 下面介绍利用平面扫描方法计算 F 的面积和周长.

先考虑一维情况. 给定数轴上 n 条线段 $L = \{l_1, l_2, \cdots, l_n\}$，其中 $l_1 = [a_1, b_1]$, $l_2 = [a_2, b_2], \cdots, l_n = [a_n, b_n]$.

计算 $l_1 \cup l_2 \cup \cdots \cup l_n$ 的长度的算法：

步 1　按 l_i 端点的横坐标分类，并存入数组 $X[1 \cdots 2n]$.

步 2　$X[0] \leftarrow X[1]$, $m \leftarrow 0$(m 是并的长度)，$c \leftarrow 0$(c 是重叠线段的次数).

步 3　for　$i = 1$　to　$2n$　do

步 4　if　$c \neq 0$　then　$m \leftarrow m + X[i] - X[i-1]$

步 5　if　$X[i]$ 是左端点　then　$c \leftarrow c + 1$

　　　else　$c \leftarrow c - 1$

显然，该算法的时间复杂性为 $O(n\text{lb}n)$. 算法中步 4，区间 $[X[i-1], X[i]]$ 长度是否加入到 m 中依赖于 c 是否为 0，$c \neq 0$ 时，则长度加入 m，$c \neq 0$ 表示扫描线碰到一区间的左端点. 该算法可以推广到二维，如图 6-1(计算垂直边的长度)所示.

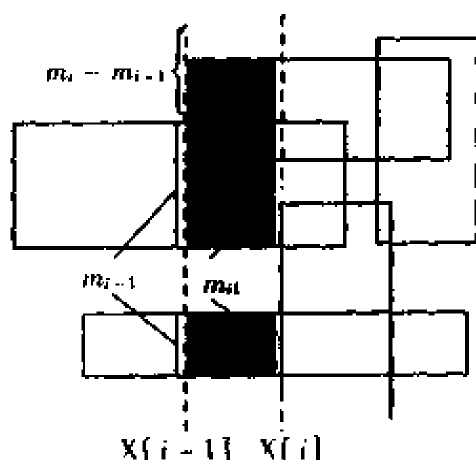


图 6-1

图 6-1 中第 i 个垂直条是 $X[i-1]$ 和 $X[i]$ 之间的平面条，该平面条的有效面积(即图中阴影部分)是 $(X[i] - X[i-1]) \times m_i$，其中 m_i 是条中一条任意垂直线和矩形并的截段的长度. 用 $O(\text{lb}n)$ 时间可以确定 m_i、m_i 是计算矩形并的面积的重要参数. 为了计算 m_i，把矩形的垂直边构造成线段树. 定义结点 v 的参数 $m[v] = [B[v], E[v]]$ 提供给 m_i 的长度，计算 $m[v]$ 的过程如下：

```
if   c[v] ≠ 0   then   m[v] ← E[v] - B[v]
else  if   v 不是叶   then   m[v] ← m[LSON[v]] + m[RSON[v]]
      else   m[v] ← 0
```

v 为线段树的根结点时，$m_i = m(v)$. 当矩形的一条垂直边插入线段树或从线段树删去时，每个结点用常数时间能保持参数 $c[v]$ 和 $m[v]$. 因此，矩形的每条垂直边用时间 $O(\text{lb}n)$ 可以插入线段树或从线段树中删去并计算 m_i. 设横坐标 $X[i]$ 处垂直边的上端点、下端点的纵坐标分别为 e_i 和 b_i，则计算矩形并的面积的算法如下.

步1　矩形垂直边按端点的 x 坐标分类，并存于数组 $X[1 \cdot\cdot 2n]$.

步2　$X[0] \leftarrow X[1], m \leftarrow 0$;

步3　构造和初始化矩形边的纵坐标的线段树 T;

步4　for $i = 1$ to $2n$ do

　　$m^* \leftarrow m[\mathrm{root}(T)]$;

　　$m \leftarrow m + m^* \times (X[i] - X[i-1])$;

　　if $X[i]$ 是左边的横坐标 then INSERT$(b_i, e_i; \mathrm{root}(T))$

　　else DELETE$(b_i, e_i; \mathrm{root}(T))$

步1耗费时间 $O(n\mathrm{lb}n)$，其它步的时间界限均不超过此量级，因此计算矩形并 F 的面积的算法的时间复杂性为 $O(n\mathrm{lb}n)$.

步1中的数组 $X[1 \cdot\cdot 2n]$ 提供事件点的进度表. 步3中 T 给出扫描线状态，它是由 $d-1$ 维推广到 d 维的关键，也是扫描方法的关键. 事实上，扫描方法把给定的 d 维问题变为 n 个 $d-1$ 维子问题的序列. 比如，$d = 3$ 时，子问题变为二维，如果能用 $O(n\mathrm{lb}n)$ 时间求解每个子问题，那么用 $O(n^2\mathrm{lb}n)$ 时间可以求解原问题. 利用四叉树和切片之间的粘合技术可以把此界限改进到 $O(n^2)$，此结果能否进一步改进，有待深入地研究.

为了计算矩形并 F 的周长，首先注意 m_i 的含义，m_i 是以 $X[i]$ 为右侧的垂直长条中垂直截线的总长度，同样 m_{i-1} 是以 $X[i-1]$ 为右侧的垂直长条中垂直截线的总长度，因此，$m_i - m_{i-1}$ 是 $X[i-1]$ 处 F 的边界上垂直边的总长度. 见图6-1所示，截线在区间 $[X[i-1], X[i]]$ 中是不变的，所以 F 的边界上垂直边的总长度在区间 $[X[i-1], X[i]]$ 内不变，该总长度只是在扫描线通过点 $X[i]$ 处发生变化. 其次要考虑垂直条 $[X[i-1], X[i]]$ 中水平边界边提供的长度. 该长度由下式计算

$$(X[i] - X[i-1]) \times \alpha_i,$$

其中 α_i 是 $[X[i-1], X[i]]$ 条中 F 的边界上水平边数目，α_i 与 m_i 相似. 图6-1中，$\alpha_i = 4$，它必为偶数. 对线段树的每个结点 v 定义3个参数 $\alpha[v]$、LBD$[v]$ 和 RBD$[v]$ 如下：

$\alpha[v]$ 是(F 的现时垂直截线) $\cap$ $[B[v], E[v]]$ 的不相交部分数的两倍；

$$\mathrm{LBD}[v] = \begin{cases} 1, & B[v] \text{ 是(}F\text{ 的现时垂直截线)} \cap (B[v], E[v]) \text{ 中一个区间的下端,} \\ 0, & B[v] \text{ 不是(}F\text{ 的现时垂直截线)} \cap (B[v], E[v]) \text{ 中一个区间的下端,} \end{cases}$$

$$\mathrm{RBD}[v] = \begin{cases} 1, & E[v] \text{ 是(}F\text{ 的现时垂直截线)} \cap (B[v], E[v]) \text{ 中一个区间的上端,} \\ 0, & E[v] \text{ 不是(}F\text{ 的现时垂直截线)} \cap (B[v], E[v]) \text{ 中一个区间的上端.} \end{cases}$$

这3个参数的初值均为0，下述过程计算该3个参数：

if $c[v] > 0$ then $\alpha[v] \leftarrow 2$; LBD$[v] \leftarrow 1$; RBD$[v] \leftarrow 1$

else $\alpha[v] \leftarrow \alpha[\mathrm{LSON}[v]] + \alpha[\mathrm{RSON}[v]] - 2\mathrm{RBD}[\mathrm{LSON}[v]]\mathrm{LBD}[\mathrm{RSON}[v]]$;

　　LBD$[v] \leftarrow$ LBD$[\mathrm{LSON}[v]]$;

RBD[v] ← RBD[RSON[v]]

$c[v]>0$时，区间[$B[v]$，$E[v]$] ⊂ F的现时垂直截线，所以$\alpha[v]=2$，LBD[v] = RBD[v] = 1，而$c[v]=0$并且F的现时垂直截线不包含跨接点E[LSON[v]] ≈ B[RSON[v]]的区间时，(F的现时垂直截线) ∩ [$B[v]$，$E[v]$]包含的项和它的两个子结点中的项的和一样多；用RBD[LSON[v]] = LBD[RSON[v]] = 1来描述该情况. 因此该计算过程是正确的. 在每个结点处，用常数时间计算参数α、LBD和RBD.

从修改F面积的算法可以得到计算F周长的算法. 当前步（垂直条）提供给周长的长度由两部分组成：如图6-2所示，在条[$X[i-1]$，$X[i]$]中的水平边，提供$\alpha_i X(X[i]-X[i-1])$；在横坐标$X[i]$处的垂直边，提供$|m_{i+1}-m_i|$. 这样，α_i的值要在后者修改之前从线段树中得到，$\alpha_i=\alpha[\text{root}(T)]$，而$m_{i+1}$是在修改之后得到的.

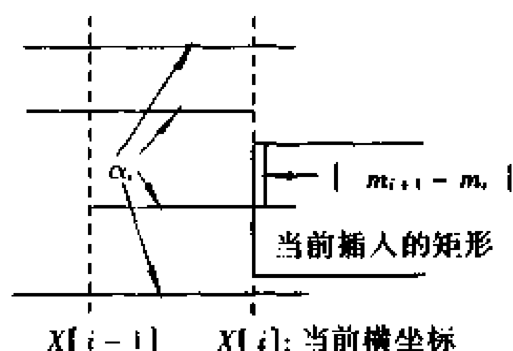


图6-2

计算F的周长的算法：

步1　矩形的垂直边按端点的x坐标分类，并存于数组$X[1\cdot\cdot 2n]$.

步2　$X[0]\leftarrow X[1]$，$m_0\leftarrow 0$，$p\leftarrow 0$.

步3　构造和初始化矩形边的纵坐标的线段树T.

步4　for　$i=1$　to　$2n$　do

　　$\alpha^*\leftarrow\alpha[\text{root}(T)]$;

　　if　$X[i]$是左边横坐标

　　　then　INSERT(b_i, e_i, root(T))

　　　else　DELETE(b_i, e_i; root(T));

　　　$m^*\leftarrow m[\text{root}(T)]$;

　　　$p\leftarrow p+\alpha^*\cdot(X[i]-X[i-1])+|m^*-m_0|$;

　　　$m_0\leftarrow m^*$

该算法用$O(n\text{lb}n)$时间可以计算F的周长.

6.3 矩形的交

本节讨论同等安置矩形的交. 显然，只有当两个同等安置矩形至少有一个公共点时，它们才相交. 先考虑一维情况，设$A=[a_1,a_2]$和$B=[a'_1,a'_2]$是两个区间，条件$A\cap B\neq\varnothing$等价于下列互斥的条件之一：

$$a_1\leqslant a_1'\leqslant a_2, \tag{6-1}$$

$$a_1'<a_1\leqslant a_2'. \tag{6-2}$$

易见，$(a_1\leqslant a_1'\leqslant a_2)\vee(a_1'<a_1\leqslant a_2')$等价于$a_1'\leqslant a_2\wedge a_1\leqslant a_2'$，即$-a_2$

$\leqslant -a_1' \wedge a_1 \leqslant a_2'$，后者是平面中点$(-a_1', a_2')$和点$(-a_2, a_1)$之间的一个优势关系$\prec$，也就是$(-a_2, a_1) \prec (-a_1', a_2')$. 这表明只要把区间变换为平面中的点，便可以把二维问题(确定n个同等安置矩形集Q中的所有相交的矩形对)与一维问题(确定n个区间的所有相交的区间对)联系起来.

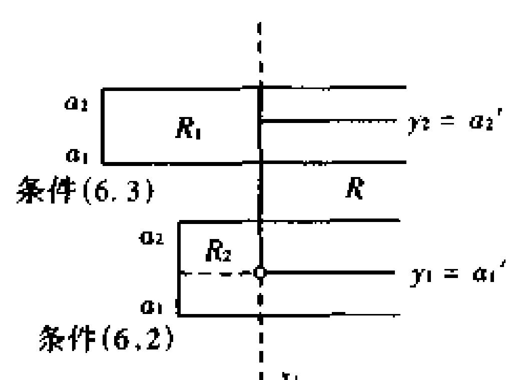


图 6-3

为了解决二维问题，采用平面扫描方法，事件点仍取为矩形的垂直边的横坐标. 由矩形和扫描线的交给出扫描线状态，与扫描线相交的矩形称为活动矩形. 在图 6-3 中，当前事件点是矩形$R = [x_1, x_2] \times [a_1', a_2']$的左侧边，扫描线横坐标$x_1$(也是$R$的$x$区间的左端)属于每个活动矩形的$x$区间. 因此，只需决定哪个活动区间关于$y$区间条件(6-1)成立或条件(6-2)成立. 利用对线段树的查找可以报告相交的矩形对，也就是说，当插入矩形的左侧边$[a_1', a_2']$时，可通过检测活动区间$[a_1, a_2]$是否$a_1' \leqslant a_1 \leqslant a_2'$，或者是否$a_1 \leqslant a_1' \leqslant a_2$来完成，当找到相交的线段对时，便找到了相交的矩形对. 产生事件点列耗费时间$O(n\text{lb}n)$，用时间$O(\text{lb}n)$插入或删去每个区间，树的查找用时间$O(\text{lb}n)$来跟踪查找路径，因此确定n个同等安置矩形的s个相交对耗费$\theta(n\text{lb}n + s)$时间.

下面介绍解决二维问题的另一种方法。

给定n个区间的集合$\{R^{(j)} = [a_1^{(j)}, a_2^{(j)}], j = 1, 2, \cdots, n$，首先定义函数$\sigma_0$: $\{R^{(1)}, R^{(2)}, \cdots, R^{(n)}\} \to E^2$，它把区间$[a_1, a_2]$映射为平面上的点$(a_1, a_2)$，该点位于第一象限的平分线的上方. 其次引入函数$\sigma_1: E^2 \to E^2$，它是平面上关于$x_2$轴的对称映射，即点$(a_1, a_2)$映射到点$(-a_1, a_2)$. 再引入函数$\sigma_2: E^2 \to E^2$，它是平面上关于第二象限的平分线$x_2 = -x_1$的对称映射，即点$(-a_1, a_2)$映射到点$(-a_2, a_1)$. 为方便起见，记复合函数$f_1 = \sigma_1\sigma_0, f_2 = \sigma_2\sigma_1\sigma_0$，并且有等价关系：

$$R^{(i)} \cap R^{(j)} \neq \varnothing \Leftrightarrow f_1(R^{(i)}) \succ f_2(R^{(j)}) \Leftrightarrow f_1(R^{(j)}) \succ f_2(R^{(i)}). \qquad (6\text{-}3)$$

图 6-4 是这种等价关系的一个例子. 显然，相交的区间对引出优势关系中的两个对.

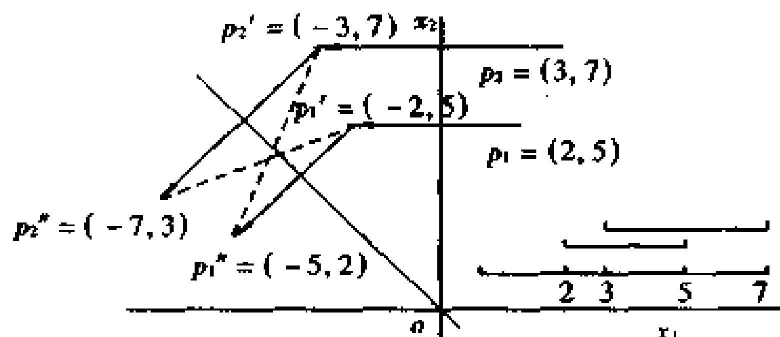


图 6-4

给定矩形 $R=[x_1,x_2]\times[y_1,y_2]$，先把 R 映射到四维空间 E^4 中的两个点 $q'(R)=(-x_1,x_2,-y_1,y_2)$ 和 $q''(R)=(-x_2,x_1,-y_2,y_1)$，其次形成 E^4 中的两个点集：$S'=\{q'(R)\mid R\in Q\}$，$S''=\{q''(R)\mid R\in Q\}$. 由(6-3)式，$R_1$、$R_2\in Q$，$R_1\cap R_2\neq\varnothing$，当且仅当 $q'(R_1)\succ q''(R_2)$ 或者 $q'(R_2)\succ q''(R_1)$. 因此，确定所有相交对问题是优势问题(给定 E^4 中两个点集 S' 和 S''，找所有对 $q'\in S'$，$q''\in S''$，使得 $q'\succ q''$) 的一个特例. 而四维优势问题存在复杂性为 $O(n\mathrm{lb}^2n+s)$ 的算法求解它，因此求解所有相交对问题的时间复杂性是 $O(n\mathrm{lb}^2n+s)$.

另外，两个矩形的位置关系有 4 种基本情况，如图 6-5 所示. 图中阴影部分是它们的交，每种基本情况产生的交都不相同. 图 6-5(a) 所示的交，两个交点位于交域的对角线上；图 6-5(b) 所示的交，两个交点位于一个矩形的右侧边上；图 6-5(d) 所示的交，两个交点位于一个矩形的左侧边上；图 6-5(c) 所示的交域由 4 个交点组成. 多个矩形的交均由上述 4 种基本情况组合而成，为了求出多个矩形的所有交，采用平面扫描方法找出交点并区分各种基本情况，便可找出所有交.

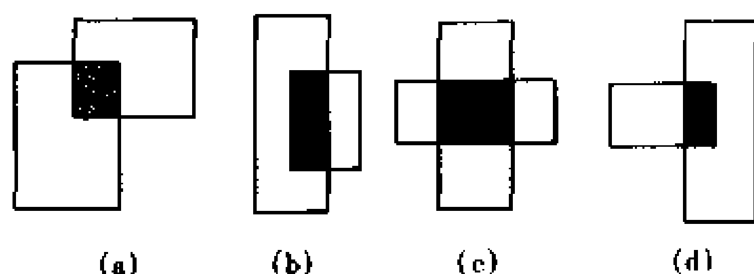


图 6-5

求矩形交的算法：

步 1　按矩形垂直边的 x 坐标分类.

步 2　扫描线从左向右移动，碰到左侧边时，检查是否有水平边落入该左侧边区间内. 如果有，则求出交点.

若有 1 个交点，扫描线右移求出另外 1 个交点，便找到 1 个交域. 如图 6-5(a) 所示情况.

若有 2 个交点，扫描线右移求出另外 2 个交点，便找到 1 个交域. 如图 6-5(c) 所示.

若有 2 个交点，扫描线右移碰到的矩形右侧边位于另一活动矩形内部，便找到 1 个交域. 如图 6-5(d) 所示.

步 3　碰到的左侧边位于另一活动矩形内部，扫描线右移求出 2 个交点，产生图 6-5(b) 所示的交. 一个矩形的左、右侧边位于另一活动矩形内部，则两个矩形呈包含关系.

7　几何体的排列

平面上的直线或三维空间中的平面的排列是S计算几何中的重要结构，它与凸壳、沃罗诺图等具有同样的重要性.本章将介绍有关排列的一些基本概念、算法和应用.

7.1　基本概念

设平面上给定 n 条直线 $s_1, s_2, \cdots, s_n$，这些直线将平面划分成凸域(或称网格或称面)、线段(两个交点之间)以及顶点(两条直线的交点)，称这种划分为**直线的排列**.图7-1示出八条直线的一种排列.

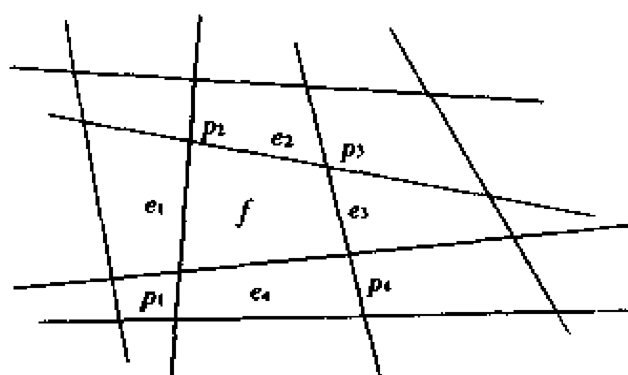


图7-1

在直线的排列中，区域(或面)称为排列的2面.直线 s_i 被其它直线分成 n 个区间，这些区间称为排列的边或者排列的1面.所有直线的交点称为排列的顶点或者排列的0-面.通常把排列的2面看成开域(不包括它们的边)，而将1-面看成是开线段(不包括端点).显然，j-面的维数是 j.这样，n 条直线的排列覆盖了整个平面，但它的2面与1面及0面之间不相交，不同的 j-面($j = 0$、或1，或2)之间也不相交.

直线的排列推广到 d 维空间 R^d 中是超平面的排列，每个超平面是满足 $a_1x_1 + \cdots + a_dx_d = a_0$ 形式的线性等式的 R^d 中点的集合，其中 x_i 表示 R^d 中的坐标，系数 a_i 是任意实数，并且 $a_1, a_2, \cdots, a_d$ 不全为0.当 $d = 2$ 时，超平面是一条直线，而 $d = 3$ 时，超平面是一个平面.

在 R^d 中给定超平面集合 N，由 N 构造的排列 $A(N)$ 是划分 R^d 为具有近邻关系且变化维的域(面).也就是说，N 中超平面将 R^d 划分成若干个凸域，这些 d 维凸域称为排列 $A(N)$ 的网格或者 d 面.在每个确定的超平面 $s_i \in N$ 上，N 中其它超平面与 s_i 的交产生 s_i 范围内的一个($d-1$)维排列，该($d-1$)维排列的网格叫做 $A(N)$ 的($d-1$)面.用同样的方法继续下去，对所有的 $j < d$，定义 $A(N)$ 的 j 面.$A(N)$ 的0面也称为顶点，1面叫做边.

如果 N 中($d=2$)每一对直线交于一个点,则称排列 $A(N)$ 是简单的,这意味着 N 中没有三条直线交于一点,并且没有两条直线是平行的.从某种意义来说,非简单的排列是退化的.

在 n 条直线组成的集合 N 上,所有简单排列 $A(N)$ 具有相同的顶点数、边数及面数.

定理 1　在 n 条直线的简单排列中,顶点数 $V=\binom{n}{2}$,边数 $E=n^2$,面数 $F=\binom{n}{2}+n+1$.并且任何非简单排列的顶点数、边数及面数都不可能超过这些数量.

依据定理 1,V、E 和 F 的数量都是 $\theta(n^2)$,所以构造平面上直线排列的算法均具有二次方复杂性.

为了设计有效的构造排列的算法,要介绍排列的一个重要组合性质,即在已有的排列中增加一条直线,该直线所能穿过的网格边的数目不会过多,这就是下面阐述的区段定理.

定理 2　设 $A(N)$ 是 n 条直线的排列,$L\in A(N)$,$A(N)$ 中与 L 相交的网格(面)的边的总数是 $O(n)$.

7.2　确定直线排列的算法

下面介绍构造直线排列的增量算法.假设已构造 $i-1$ 条直线的排列 $A(i-1)$,插入第 i 条直线 L_i 之后,要求构造排列 $A(i)$.为此,需要求出 $A(i-1)$ 与 L_i 的所有交点.首先用常数时间找到 L_i 与 $A(i-1)$ 中的任意一条直线的交点 x,如图 7-2 所示,$x=L_i\cap L_2$.然后由 x 向前沿 L_i 的区段 $Z(L_i)$ 的每个面的边按顺时针方向前进,即图 7-2 中用曲线表示的路径,直至遇到 L_i 与另一条直线相交并求出交点.重复此

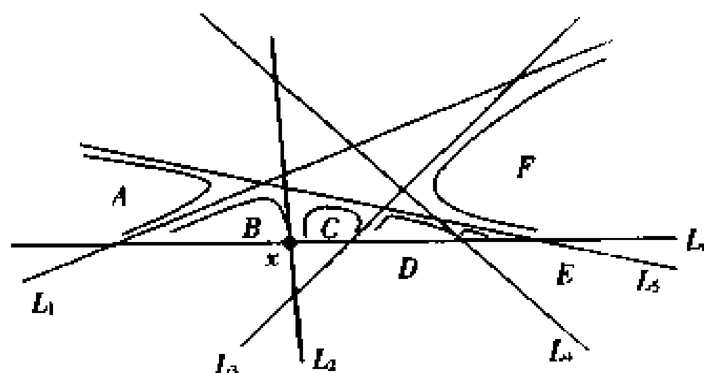


图 7-2

过程,当遇到区段的一条无穷边时前进终止.在图 7-2 中,从 x 出发经 C 的三条边遇到 L_3 和 L_i 相交求出交点之后,重复该过程,即经过 D 的三条边、E 的两条边和 F 的三条边,最后进入 F 的一条无穷左边界.这样便求出 L_i 与 L_3、L_4、L_5 的交点,终止该过程.同理,从 x 向后沿 B 的边界按逆时针方向经过 B 的三条边,求出 L_1 与 L_i 的交

点之后，再沿 A 的边界按逆时针方向经过 A 的一条边，最后进入 A 的一条无穷右边界，终止该过程.经过每个面的边界耗费常数时间，插入一条直线的总耗费取决于区段的复杂性，由定理 2 知，该复杂性是 $O(n)$. 注意，如何使用排列的结构来避免分类，在找到与 L_i 相交的所有交点之后，耗费 $O(n)$ 时间可以将 $A(i-1)$ 修改为 $A(i)$.因此整个构造过程需要 $O(n^2)$ 时间.

构造直线排列的增量算法：

```
begin
    构造 A(0)，空排列的一个数据结构.
    for  i = 1,2,…,n  do
       将直线 L_i 插入A(i - 1)，过程如下：
          寻找 L_i 和A(i - 1) 中某直线的交点 x.
          由 x 出发沿 Z(L_i) 中网格向前移动，
          由 x 出发沿 Z(L_i) 中网格向后移动，
       修改 A(i - 1) 成为 A(i)
end
```

定理 3　平面上 n 条直线的排列可以在 $\theta(n^2)$ 时间及空间内构造.

7.3　应　　用

排列有许多应用，本节仅介绍排列在删去隐藏面及特征图中的应用.

1. 删去隐藏面

现今人们利用计算机制作电视广告或者电影特技，在这些应用中最频繁使用的是删去隐藏面.例如将多种彩色多边形拼接成所需要的各种图案，如果这种拼接允许重叠，那么就要删去被遮盖的部分.

设输入的多边形顶点数目为 n，并且一组多边形遮盖另一组多边形，如图 7-3 所示，这时至少产生 $n^2/16$ 个交点，在删去由这些交点组成的某些多边形之后，才能显示出两组多边形重叠的场景.因此在最坏情况下最佳算法的时间复杂性不低于 $O(n^2)$.现已有处理该问题的许多算法，它们的复杂性为 $O(n^2 \mathrm{lb} n)$，比该问题的复杂性下界高 $\mathrm{lb} n$ 倍.Mckenna 利用排列方法已设计出最坏情况复杂性为 $O(n^2)$ 的算法.下面简要介绍这种算法.

假设多边形的空间不相互穿透，即它们可以有重叠的部分和重叠的边界，但它们的内部是分离的，另外，假设视点离多边形无穷远，使得所有视线是平行的，并且不处理透视的情况.不失一般性，设眼睛在$(0,0,+\infty)$处，即在 z 轴正向无穷远处，这样视野平面是 xy 平面，$z=0$.在所有多边形下面放置一个充分大的背景多边形 B，使得所有视线都碰到 B.问题是要确定由给定观察点可见到的场景.另外，一般问题可以归结到这个问题.

首先把输入的凸多边形的每条边投影到 xy 平面(只要删去端点的 z 坐标)，这称为**正交投影**.然后在 xy 平面上将投影线延伸成直线，这样便构成 xy 平面上 n 条直线的一种排列 $A(n)$，由 7.2 节知，在 $O(n^2)$ 时间内可以构造 $A(n)$.现要确定

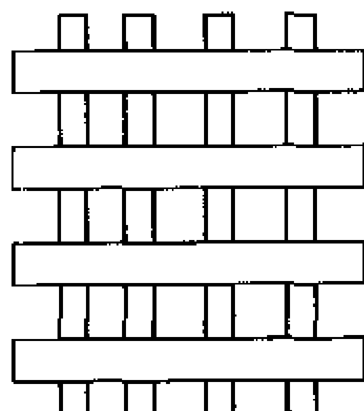


图 7-3

$A(n)$ 的一个网格，它们是由多边形集合中位置最高的多边形投影到 xy 平面上形成的，该多边形离眼睛最近、而离 xy 平面最远．由于每个凸多边形投影到 xy 平面上只产生一个网格，所以可以对网格着与形成该网格的多边形相同的颜色．

一种简单的算法需要 $O(n^3)$ 时间，这是由于 $A(n)$ 有 $O(n^2)$ 个网格，对 $O(n^2)$ 个网格中的每个网格，计算 $O(n)$ 个多边形的每一个多边形的高度，而要求每个网格仅耗费常数时间．

Mckenna 算法使用了排列的拓扑扫描，他推广了 Edelsbrunner 和 Guibas 提出的平面扫描方法，这种方法不是扫描排列上面的一条垂直线，而是扫描一条垂直的虚设线 L，L 是与 $A(n)$ 中的每条直线仅相交一次的曲线．在交点处 L 从下向上穿过 $A(n)$ 中的直线．取 L 为曲线的优点是，在优先队列查找中不必耗费 $O(\text{lb}\,n)$ 时间确定哪个点是下一个要扫描的点．也就是说，可以保持可扫描顶点的一个无序的聚集：L 穿过的那些顶点与相邻的两条边关联．图 7-4 中顶点 v 是可扫描的，因为 L 穿过的网格 C 的两条相邻的边与 v 是关联的．

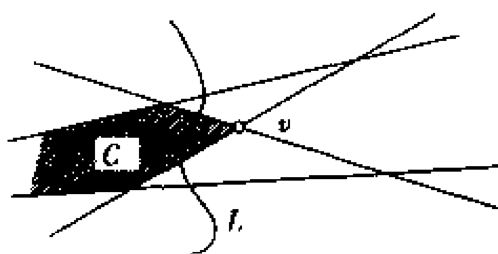


图 7-4

除了确定的排列之外，算法保持的数据结构包括激活的网格表和 L 穿过的边(比如图 7-4 中阴影网格）的表，并且对于每个激活的网格 C，其投影包含 xy 平面中网格 C 的所有多边形的表，将这些多边形对 z 深度进行分类．注意，只对激活的网格保持这些表．由于 L 穿过所有 n 条直线，所以总是有 $n+1$ 个网格．这些表能够提供足够的信息以确定 $A(n)$ 的每个网格的最前面的多边形．当扫描一个顶点时，旧的网格消亡而新的网格被激活，但它们包含的多边形的表或者相同或者几乎相同．为了使 $A(n)$ 的所有网格上面每个网格的修改只耗费常数时间，可以利用这种相关性来传递穿过已扫描过的顶点的足够信息．这种删去隐藏面的算法在最坏情况下的复杂性是 $O(n^2)$．

实际应用时这个算法不是最好的，因为它总是需要 $O(n^2)$ 时间和空间．在许多实际问题中，重叠部分没有图 7-3 那么复杂，因此要设计出对输出场景复杂性敏感的算法，这种算法称为输出规模敏感的隐藏面算法，这是当前的一个研究课题．

2. 特征图

计算机视觉中的特征图(aspect graphs) 的概念是为辅助图像识别而提出的，其思想是存储一个物体可以提供给观察者的全部特有的视野，然后把这些视野与实际上所看到的视野进行比较．对于一个多面体，利用组合的等价性确定特有的视野：如果图像有相同的组合结构，也就是说，视野平面上多面体的可视面投影所产生的已标记的平面图是相同的，那么从两个视点所看到多面体的特征相同．可视空

间划分(VSP)是一种把对象外部所有空间划分成连通域或者特征不变的网格. 最后,特征图是VSP的对偶,每个域对应一个结点,并且给面的连通域分配一条弧.

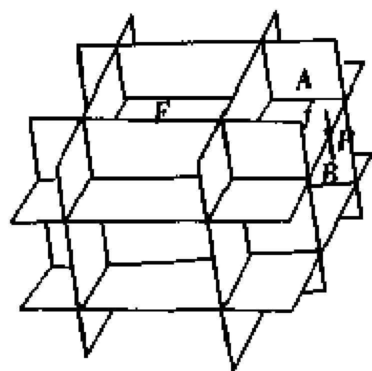


图 7-5

排列为理解凸多面体的VSP(或者对偶的特征图)提供了一种几何结构. 对于多面体 P,其VSP的确是由包含 P 的面所在的平面形成的排列. 例如,考虑图7-5所示立方体的排列,它划分空间成26个无界域,其中6个是基于立方体面的矩形柱体,8个是与顶点关联的并位于顶角的柱体,以及12个楔,每个楔与立方体的一条边关联. 考虑从网格 A 移动的点 p,从 p 观察立方体,穿过排列的面 f,进入相邻的网格 B,如图7-5中解释的那样. 假设从网格 A 来观察,排列面 f 所位于的那个平面的立方体面 F 是可视的,那么当 p 在 f 上时,就是从边上看 F,而当 p 移动到网格 B 时,F 就不可视了. 因此,f 的确表示特征的转变.

由定理7-3的推广能够得到 n 个顶点的凸多面体的VSP有规模 $O(n^3)$,并且可以在 $O(n^3)$ 时间内构造. 依据VSP的表示,特征图是有效的.

8 算法的运动规划

S计算几何中的一些研究问题来源于机器人学领域. 这些问题称为运动规划或者更准确地称为算法的运动规划. 本章研究该领域中的某些问题及其求解它们的算法.

假设二维和三维空间中存在多个障碍物,这些障碍物是多边形或者多面体. 另外,一个可运动的物体从空间中的点 s 移动到另一点 t,运动的物体碰到障碍物时必须绕过它. 这里可以提出许多问题,例如,求从 s 至 t 的最短路径;运动物体的形状是点、一条线段、一个凸多边形、一个圆或者任意多边形,求从 s 至 t 的路径等等. 物体在运动过程中要避免与所有障碍物之间的碰撞,也就是说,物体边缘上的任一点 a 与障碍物的一个内点重合时,就发生碰撞,点 a 沿障碍物边界的滑动是允许的. 没有碰撞的路径称为自由路径. 本章主要讨论三类问题:

(1) **判定问题** 运动物体(即机器人)R 能够从 s 移动到 t 吗?即从 s 至 t 是否存在一条自由路径.

(2) **构造路径** 寻找机器人 R 从 s 移动到 t 的一条自由路径.

(3) **最短路径** 寻找机器人 R 从 s 移动到 t 的一条最短自由路径.

另外,当 R 具有不同几何外形时,可以再次提出上述三个问题. 第1个问题比第2个问题容易(可以举出例子来说明这一点),而第2个问题又比第3个问题简单. 第3个问题中的"最短"的含义,一般是指自由路径长度最短. 但进一步研究发

现"最短"与 R 的外形也存在一定关系.如果 R 是一个圆,那么"最短"的含义是清楚的;而 R 是一条可以旋转的线段时,理解"最短"的含义就不那么简单了,这时先给出几个不同的定义,在此定义下求最短路径.

下面仅考虑几个最短路径问题:首先,机器人是一个点(8.1 节).然后,研究两个容易理解的运动规划问题:移动圆盘(8.2 节)和平移一个凸多边形(8.3 节).

8.1　最短路径

本节假设障碍物是多个离散的多边形,而且多边形顶点总数为 n,给定起点 s 和终点 t,s 和 t 不在任一障碍物多边形内部,问题是寻求 s 和 t 之间的最短路径.解决这个问题的一种方法,其思路是先求可视图,然后由可视图寻找最短路径.

1. 可视图及其构造

一组多边形的可视图 $G=(V,E)$,其中结点 $v(\in V)$ 对应于多边形的顶点,而边 $e(\in E)$ 对应于可以互相"看见"的顶点对($p_{ki}, p_{(k+1)j}$).注意,边 e 可能与多边形边重叠,这样,多边形的边也在可视图中.

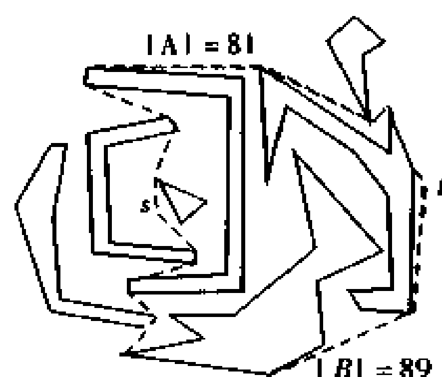


图 8-1

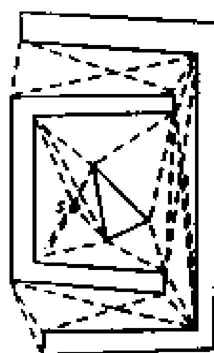


图 8-2

从 s 至 t 的最短路径是由线段组成的一条折线,见图 8-1 所示,折线的两端点或者是 s、t,或者是多边形的顶点.也就是说,多边形的顶点可以取为起点和终点.这样,最短路径可以看成是障碍多边形顶点的序列,或者最短路径是由可视图(图 8-2)中边组成的序列.基于这个观察,有下面的结论.

定理1　最短路径是障碍多边形顶点可视图的一条子路径.

由于可视图是有穷的,依据该定理,从 s 至 t 的最短路径必由一有穷路径集合中可以搜索得到.但该路径集合可以表示为多级多叉树结构,因此路径的数目可能以 n 的指数增长.为了获得有效的最短路径算法,先考虑可视图的构造.

构造一组多边形可视图的边几乎与寻找多边形对角线相同,唯一差别是:多个多边形与一个多边形,外部可视与内部可视.一种算法是对于不同多边形的每个顶点对 p_{ki} 和 $p_{(k+1)j}$,检查 $\overline{p_{ki}p_{(k+1)j}}$ 是否与多边形边相交,如果都不相交,则 $\overline{p_{ki}p_{(k+1)j}}$ 是可视图的一条边,否则就不是可视图的边.由于顶点对数目可以达到 $O(n^2)$,而多边形边数为 $O(n)$,所以该算法的复杂性为 $O(n^3)$.另外,可视图边的数目为

$O(n^2)$，因此，$O(n^2)$是寻找可视图的任意算法的一个下界．利用第7章介绍的排列方法导致一个最佳算法，时间复杂性为$O(n^2)$．经过长期的研究之后，Ghosh和Mount找到了一个关于输出规模敏感的算法，复杂性是$O(n\text{lb}n+|E|)$，当然，$|E|=O(n^2)$，但是多数情况下，$|E|$比$\binom{n}{2}$小得多．

2. 戴克斯徒拉算法

假设已构造出一组多边形的可视图，在该图中如何寻找出一条最短路径，这是图论中所研究的一个问题：寻找加权图中的一条最短路径．加权是指边的长度，用欧几里德距离度量边的长度．戴克斯徒拉(Dijkstra)于1959年提出了解决这个问题的一个算法．介绍该算法之前先通过一个例子看其主要思想．

在图8-2中，设想可视边都是由红色液体在其内流动的细管组成，并且红色液体由s出发以相等的速率沿管道(可视图边)流向还是空的管道，随着时间的推移，被染红的管道越来越多，路径也越来越长，直到某时刻g，终点t被着色．此时由s至t的一条最短路径被红色管道显示出来．

戴克斯徒拉算法的思想是模拟上述着色过程．设已知图G中最接近于始点s的m个结点以及从始点s到这些结点中每一个结点的最短路(从s到其本身的最短路是零路，即没有弧的路，其长度为0)，对始点s和这m个结点着色．然后，最接近于s的第$m+1$个结点可如下求之：

对于每一个未着色的结点y，考虑所有已着色的结点x，把弧(x,y)接在从s到x的最短路后面，这样就得到从s到y的m条不同路．从这m条路中选出最短的路，它就是从s到y的最短路．相应的y点就是最接近于s的第$m+1$个结点．因为所有弧的长度都是非负值，所以从s到最接近于s的第$m+1$个结点的最短路必然只使用已着色的结点作为中间结点．

从$m=0$开始，将这个过程重复进行下去，直至求得从s到t的最短路为止．

戴克斯徒拉最短路算法：

步1　开始所有弧和结点都未着色．对每个结点x指定一个数$d(x)$，$d(x)$表示从s到x的最短路的长度(中间结点均已着色)．开始时，令$d(s)=0$，$d(x)=\infty$(对所有$x\neq s$)．y表示已着色的最后一个结点．对始点s着色，令$y=s$．$T\leftarrow\varnothing$．

步2　对于每个未着色结点x，重新定义$d(x)$如下：

$$d(x)=\min\{d(x),d(y)+a(y,x)\}.$$

其中$a(y,x)$表示弧(y,x)的长度．对于所有未着色结点x，如$d(x)=\infty$，则算法终止．因为此时从s到任一未着色的结点都没有路．否则，对具有$d(x)$最小值的未着色结点x进行着色．同时把弧(y,x)着色(指向结点x的弧只有一条被着色)，即$T\leftarrow$弧(y,x)．令$y=x$．

步3　如果结点t已着色，则算法终止．这时已找到一条从s到t的最短路，如果t未着色，则转步2．

注意，已着色的弧不能构成一个圈，而是构成一个根在s的树形图T，此树形图称为最短路树形图．若x是最短路树形图中的任一结点，则从s到x的唯一的一条路是从s到x的最短路．

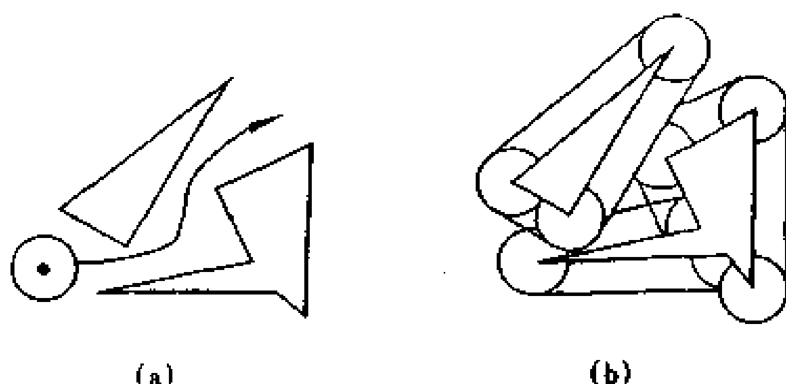
这个算法可以看成是根在始点 s 的树形图的生长过程.一旦到达终点 t,生长过程就停止.由于已假设多边形组有 n 个顶点,因此图 G 有 n 个结点,并且至多有 $\binom{n}{2}$ 条边.戴克斯徒拉算法每循环一次,处理1个结点,向 T 中加入1条边,所以戴克斯徒拉算法至多需要循环 n 次.每次循环中要检验的候选边的数目大约是 $O(n^2)$,因为可视图可能有平方阶条边.这样,粗略分析戴克斯徒拉算法的时间复杂性是 $O(n^3)$.如果注意到每次循环中不必重新检验这些边,那么在 $O(n^2)$ 时间内可以运行戴克斯徒拉算法,加之可视图的构造需要 $O(n^3)$ 时间,故而有下面的定理.

定理 2 在有 n 个顶点的多边形组中移动一个点,其最短路径可以在 $O(n^2)$ 时间和空间内找到.

8.2 移动圆盘

本节讨论运动规划的算法,目的是寻找任意路径(如果一条路径存在的话),而不是最短路径.

假设机器人 R 是一个中心在 s 的圆盘,移动 R 一段距离之后,使 R 的中心位于 t,并且 R 不穿透任何障碍物.仍设障碍物是分离的多边形.图8-3(a)中所示路径不是一条可通行的路径,因为机器人太宽,不能通过所指示的通道.下面的方法用于判定圆盘机器人 R 能否通过指定通道是非常有效的.设圆盘 R 的中心为 r,那么 r 不能离任意特定多边形 P 太近,也就是说,圆盘 R 在移动的过程中不可能移动到距 P 边的距离小于圆盘半径 ρ 的位置.这样,考虑一个扩展的障碍物 P'(P 向外扩充距离 ρ)以便移动点 r,见图8-3(b)所示机器人 R 已收缩成一个点,并且把障碍物向外扩充 ρ,因此,上述问题简化为8.1节中的问题.



(a) (b)

图 8-3

在图8-3(b)中,圆盘环绕 P 的边界移动时,圆盘中心 r 留下的轨迹便是 P' 的边界.如果 r 位于 P' 的外部,那么 R 与 P 不相交;否则就相交.显然,障碍物增大之后通道重叠,因此 R 无法通过图8-3(a)中所指示的路径.上面描述 P' 的方式不清晰,下面介绍使用两个点集的闵可夫斯基(Minkowski)和的概念来描述 P' 将更明确.

给定平面上两个点集 S_1 和 S_2，如果在平面上建立一个坐标系，那么便可以把点看成该坐标系中的矢量. 定义 S_1 与 S_2 的和：$S_1 + S_2 = \{x + y \mid x \in S_1, y \in S_2\}$，其中 $x + y$ 是点 x 与点 y 的矢量和，这称为 S_1 与 S_2 的闵可夫斯基和. 另外，点 x 与集合 S_2 的闵可夫斯基和为 $x + S_2 = \{x + y \mid y \in S_2\}$，因此，对于每个 $x \in S_1$，$S_1 + S_2 = \bigcup_{x \in S_1}(x + S_2)$ 是 S_2 的复制的并. 现设 S_1 是一个多边形 P，并且 S_2 是中心在原点的圆盘 R，那么对于所有 $x \in P$，可以把 $P + R$ 看成是依据 x 转换的 R 的复制. 由于 R 的中心在原点，$x + R$ 的中心将在 x，因此 $P + R$ 相当于放置中心在 P 的每个点之上的 R 的复制，故 $P + R$ 是 P 的扩展域 P'.

在图 8-3(b) 中，考查三角形的扩展. 当 x 是三角形的三个顶点时，在这些顶点处放置圆盘 R 的中心，这就完成了顶点处圆盘的复制工作. 而当 x 位于三角形边界时，圆盘 R 的周边位于两端点圆盘的切线上，这就相当于圆盘中心沿三角形边界滑动时，R 的周边产生的轨迹. x 位于三角形内部时，$x + R$ 位于 P' 的内部.

给定一组离散的多边形和半径为 ρ 的一个圆盘，圆盘 R 的中心 r 放在始点 s 上，t 为终点，寻找由 s 至 t 的一条路径，使得圆盘 R 沿该路径从 s 可以移动到 t（即中心 r 落在 t 上）. 下面叙述解决该问题的一种算法的粗略步骤.

寻找圆盘 R 从 s 移动到 t 的路径的算法：

步 1　利用闵可夫斯基和及圆盘 R 扩大每个障碍物.

步 2　构造已扩大障碍物的并.

步 3　如果终点 t 与始点 s 位于平面上相同的部分之中，则由 s 至 t 有一条路径存在，并且通过改进可视图使之包含圆的弧，可以找到最短路径；否则，s 与 t 之间不存在路径.

该算法的时间复杂性为 $O(n^2 \text{lb} n)$.

8.3 平移凸多边形

当机器人 R 是一个凸多边形时，机器人可以采取旋转的运动方式从一个位置移到另一个位置，但本节将机器人的运动方式限制为平移. 显然，凸多边形比圆盘复杂些，但利用闵可夫斯基和扩展障碍物的思想仍然有效. 先介绍一个简单例子，由这个例子可以说明算法的基本思想.

设机器人 R 是一个正方形，以该正方形左下角 r 为参考点，多边形 P 是如图 8-4 所示的五边形. 当 R 围绕 P 的边界 ∂P 移动时，r 描划出 P' 的边界，规定 r 不能穿透平面域上的一个已扩展的障碍物. 这里的情况与 8.2 节不同，P 的扩展是通过 r 点的运动来实现的. 如图 8-4 所示，点 r 沿边 e_1 运动时，不必扩展 P，也就是说，e_1 是 P 和 P' 的共同边界；R 沿边 e_2 运动时，以 R 的水平宽度为 P' 的边界；R 沿 e_5 运动时，以 R 的垂直高度为 P' 的边界；R 沿 e_3、e_4 运动及在凹点的情况，P' 如图 8-4 中虚线所示.

在 R 是一个圆盘的情况下，$P' = P + R$，其中"+"是闵可夫斯基和，但这个关系式在图 8-4 中显然不成立，例如，在 P 的边 e_1 处 $P + R$ 向外凸出，但 P' 不是这样.

这里的计算要通过参考点 r 用 R 的反射取 P 的闵可夫斯基和.因为对于闵可夫斯基和形式来说 r 是原点,R 的这种反射形式只不过是 $-R$,即求反 R 的每个点.因此图 8-4 中的 P' 是 $P+(-R)=P-R$.而圆盘关于其圆心是中心对称的,$R=-R$,因此这个新的形式与上一节的表示是一致的.由于闵可夫斯基减法就是反射域的加法,故仍将称它为闵可夫斯基加法.

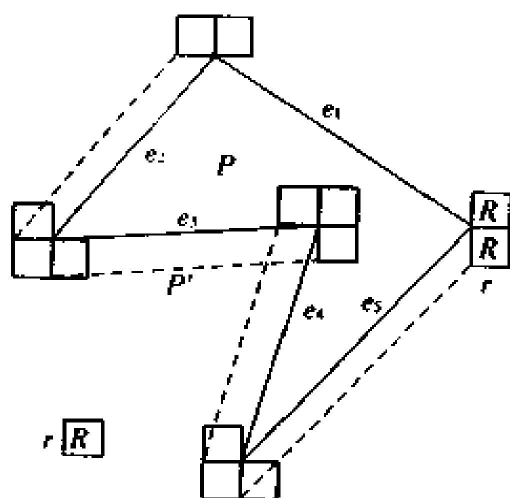


图 8-4

定理 3 设 R 是包含原点(参考点)的一个域(机器人),而 P 是一个障碍物,那么在下述意义下域 $P'=P-R$ 不能被 r 穿透:

1° 如果平移 R 使得 r 进入 P' 的内部,那么 R 穿透 P.

2° 如果平移 R 使得 r 位于 $\partial P'$ 上,则 ∂R 与 ∂P 相切.

3° 如果平移 R 使得 r 在 P' 的外部,则 $R\cap P=\varnothing$.

实际上,R 和 P 可以不是凸的,也不必是多边形.但本节仍设两者是多边形,并且 R 是凸的.

下面给出凸多边形 R 规划运动的算法概要.

设障碍物 $P_1,P_2,\cdots,P_m$ 共有 n 个顶点,算法描述如下:

步 1　增大所有的障碍物:$P'_i=P_i-R,i=1,2,\cdots,m$.

步 2　构造它们的并 $P'=\bigcup_i P'_i,i=1,2,\cdots,m$.

步 3　寻找包含 s 和 t 的连通域,设为 K.

步 4　在 K 中寻找 s 和 t 之间的一条路径.

图 8-5 所示是一个平移凸多边形 R 的例子,机器人 R 是一个四边形,其内部的

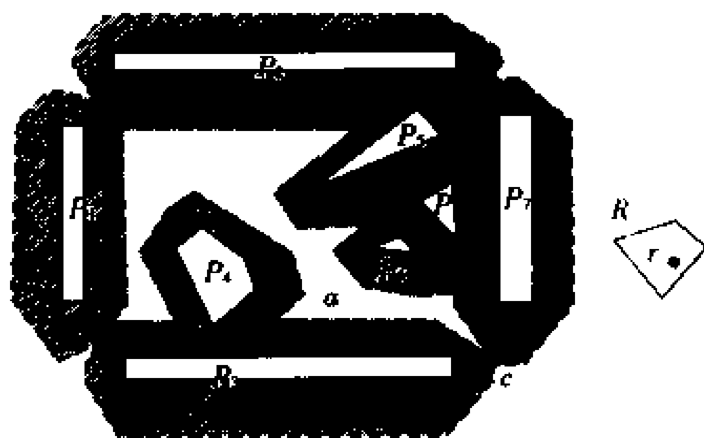


图 8-5

任意一点选作参考点 r,图中有 7 个障碍物 $P_1,P_2,\cdots,P_7$.执行该算法之后,得到三

个连通域 a、b 和 c. 只要始点 s 和终点 t 位于同一连通域内，机器人 R 就可以由 s 移动到 t. 因此，设计机器人的路径问题简化为在同一连通域内寻找参考点的一条路径.

定理 4　设多边形障碍物顶点总数为 n，机器人 R 为凸多边形，寻找 R 由起点 s 到终点 t 的平移路径在 $O(n\mathrm{lb}n)$ 时间内可以完成. 如果 R 有 k 个顶点，则时间复杂性是 $O(kn\mathrm{lb}(kn))$.

Kedem 和 Sharir(1990) 证明了上述定理.

参 考 文 献

1　Chazelle B. Computational geometry: a retrospective. In: Du DZ, Hwang F, ed. Computing in Euclidean geometry. Singapore: World Scientific Publishing Co. Pte. Ltd., 1995, 4, 22～46.
2　Goodrich M T. Parallel algorithms in geometry, In: Goodman JE. and O'Rourke J. ed. Handbook of Discrete and Computational Geometry. New York: CRC Press LLC, 1997.
3　Mulmuley K. Computational geometry: an introduction through randomized algorithms. Englewood Cliffs: Prentice Hall, 1994.
4　Mulmuley K, Schwarzkopt O. Randomized algorithms, In: Goodman JE and O'Rourke J ed. Handbook of discrete and computational geometry. New York: CRC Press LLC, 1997.
5　O'Rourke J. Computational geometry in C. Cambridge: Cambridge Univ Press, 1994(第一版). 1998(第二版).
6　Preparata F P, Shamos M I. Computational geometry: an introduction. New York: Springer-Verlag, 1988.
7　Shamos M I. Problems in computational geometry, unpublished manuscript, 1975.
8　Mac Gregor Smith J, Winter P. Computational geometry and topological network design, In: Du DZ and Hwang F ed. Computing in Euclidean geometry. Singapore: World Scientific Publish Co, 1995. 351～451
9　Berg M D, Kreveld M V, Overmars M, Schwarzkopf O. Compatational geometry: algorithms and applications. Berlin: Springer-Verlag 1997.
10　Boissonnat J D and Yvinec M. Algorithmic geometry. London: Cambridge University. Press, 1998.
11　Chin F, Snoeyink J, Wang C A. Finding the medial axis of a simple polygon in linear time. Discrete and Computational Geometry, 1999(21): 405～420
12　周培德. 计算几何——算法设计与分析. 北京: 清华大学出版社, 广西科学技术出版社, 2000.

·计算机数学卷·

第22篇

代数编码

编　者　杨义先　孙　伟　钮心忻
审校者　胡正名

目　录

引　言

代数编码又名纠错编码、差错控制编码或信道编码，此种编码所使用的主要数学工具是代数. 在通信工程应用中的主要目的是发现并自动纠正数据信息在传输过程中出现的差错. 它还常用于诸如磁记录和介质记录(比如计算机磁盘等)等信息存储的差错检测和纠正. 目前此种编码技术已经广泛地应用于深空通信、军事通信、数据通信、信息检索、数据及计算机存储系统、大规模和超大规模集成电路设计等领域.

1948 年香农(C. Shannon)在他的著名经典论文"通信的数学理论"中指出：在有扰信道中，当信息传输速率低于信道容量时，通过某种编译码方法随着码长的增加能使误码率任意小，从而达到可靠通信之目的. 但遗憾的是，香农并未确定性地找到能使误码率任意小的编码方法. 过去 50 年来，全世界的编码学家们一直致力于寻找使误码率尽可能小的编码方法，从而导致了代数编码理论的诞生. 特别是汉明(Hamming)等在 20 世纪 50 年代初期开创的设计好码和有效译码方法引起了许多学者和数学家的广泛关注. 自 20 世纪 70 年代以来，由于大规模集成电路技术和卫星、通信技术的发展，使得代数编码研究的重点转向工程应用. 最近几年纠错编码研究又取得了重要突破，一种新型的编码(turbo 码)由于其接近理论极限的性能，掀起了编码研究的又一个新高潮. 由此可见，代数编码这门学科是极富生命力的学科.

1　线性分组码

1.1　基本概念

1.1.1　线性分组码的定义

有限域 GF(2) 上的所有 n 维向量在对应元素模 2 加的运算下构成一个 n 维向量空间，此向量空间的 k 维子空间就称为(n,k) **线性分组码**(简称为**线性码**)，其中 n 称为**码长**，k 称为**信息位数**. 此子空间中的每个向量称为分组码的一个码字，共有2^k 个码字. 任意两个码字的对应分量模 2 相加之后得到的向量也是一个码字.

例 1　由偶数个"1"和若干个"0"组成的 n 维向量的集合形成一个$(n, n-1)$线性分组码. 由全"0"向量和全"1"向量两个向量组成的集合形成一个$(n,1)$线性分组码.

(1) **生成矩阵和校验矩阵**　设 $g_0 = (g_{00}, \cdots, g_{0,n-1}), \cdots, g_{k-1} = (g_{k-1,0}, \cdots,$

$g_{k-1,n-1}$)是(n,k)线性码C中的k个相互独立的码字.于是,对任意一个码字$\boldsymbol{v}=(v_0,\cdots,v_{n-1})$,都存在唯一向量$\boldsymbol{u}=(u_0,\cdots,u_{k-1})$,$u_i=0$或1,使得$\boldsymbol{v}=u_0\boldsymbol{g}_0+u_1\boldsymbol{g}_1+\cdots+u_{k-1}\boldsymbol{g}_{k-1}$.将向量$\boldsymbol{g}_i$,$0\leqslant i\leqslant k-1$,按行排列成的$k\times n$阶矩阵,称为$(n,k)$线性码的生成矩阵,记为$\boldsymbol{G}$.一个线性码可能有多个不同的生成矩阵.每个码字都是由某个k维向量与生成矩阵相乘而得到的.

如果线性码C的生成矩阵形如$\boldsymbol{G}=[\boldsymbol{P}\mid \boldsymbol{I}_k]$,那么称此码为**系统码**,它的前面$n-k$位称为**冗余校验部分**,后面$k$位称为**消息部分**.称矩阵$\boldsymbol{H}=[\boldsymbol{I}_{n-k}\mid \boldsymbol{P}^{\mathrm{T}}]$为码$C$的**一致校验矩阵**,此处$\boldsymbol{I}_{n-k}$表示$n-k$阶单位矩阵,$\boldsymbol{P}^{\mathrm{T}}$表示矩阵$\boldsymbol{P}$的转置.

(n,k)线性码C的生成矩阵$\boldsymbol{G}$与一致校验矩阵是相互正交的,即$\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$,换句话说,$n$长向量$\boldsymbol{v}$是一个码字,其充要条件是$\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$.

如果一个线性码C_1的生成矩阵是另一个线性码C_2的校验矩阵,则称C_1是C_2的**对偶码**(C_2当然也是C_1的对偶码).

例2　以下两个矩阵$\boldsymbol{G}$和$\boldsymbol{H}$形成一对相互正交的矩阵,它们既可以分别作为某个线性码C的生成矩阵和一致校验矩阵,又可以作为某组对偶码C_1、C_2的生成矩阵.

$$\boldsymbol{G}=\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\1&1&1&0&0&1&0\\1&0&1&0&0&0&1\end{bmatrix},\quad \boldsymbol{H}=\begin{bmatrix}1&0&0&1&0&1&1\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}.$$

(2) **汉明距离**　线性码C中每个码字所包含"1"的个数称为此**码字的汉明重量**;某两个码字对应分量模2加以后得到的向量的汉明重量称为**这两个码字间的汉明距离**;码C中任意两个码字之间汉明距离的最小值称为**C的最小汉明距离**(或简称为最小距离).

线性码的最小距离等于非全零码字的最小汉明重量.

如果线性码C的一致校验矩阵为$\boldsymbol{H}$,则C的最小距离等于$\boldsymbol{H}$中和值为0向量的最小列数.

(3) **最小距离译码原则**　是将接收向量译码为与其汉明距离最短的那个码字.

设C是任意一个码长为n的线性码,定义它的**支撑**$\chi(C)$为

$$\chi(C)=\{i\mid 存在(c_1,c_2,\cdots,c_n)\in C 使得 c_i\neq 0\}.$$

现在设C是一个(n,k)线性码,对于任意r,$1\leqslant r\leqslant k$,C的**第r个广义汉明重量**定义为

$$d_r(C)=\min\{\mid\chi(D)\mid,D 是 C 的 r 维子码\}.$$

显然,$d_1(C)$就是线性码C的最小汉明重量.码C的所有广义汉明重量的集合$\{d_1(C),d_2(C),\cdots,d_k(C)\}$称为**$C$的重量谱系**.

1.1.2　检错纠错能力

当码字$\boldsymbol{v}=(v_0,v_1,\cdots,v_{n-1})$通过有扰信道传输时,它会受到噪声$\boldsymbol{e}=(e_0,e_1,$

$\cdots, e_{n-1}$)(又称为错误图样或错误矢量)的干扰,使得接受端收到的消息变成 $r = (r_0, r_1, \cdots, r_{n-1}) = v + e$.如果接受端发现 $rH^T \neq 0$(即 r 不再是一个码字),那么可以断定出现干扰;但是如果 $rH^T = 0$(即 r 仍然是一个码字),那么 r 接受端便无法确定是否出现过干扰.此时便出现了一个不可检测错误图样,当然也就出现一个译码错误.称值 $s = rH^T = eH^T$ 为 r 的**伴随式**.

最小距离为 d 的(n,k)线性码能够检测所有 $d-1$ 个或更少个错误的错误图样.共有 $2^n - 2^k$ 个错误图样是可检测的.

设 C 是一个(n,k)线性码.令 $A_i, 0 \leqslant i \leqslant n$,表示码 C 中重量为 i 的码字个数,称$\{A_0, A_1, \cdots, A_n\}$ 为码 C 的**重量分布**.如果二元对称信道的转移概率为 p,那么码 C 的不可检测错误概率 $P_u(E)$ 等于

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}. \tag{1-1}$$

最小距离为 d 的线性码能够纠正$\lceil d-1)/2\rceil$个或更少个错误的错误图样,其中$\lceil X\rceil$表示不大于 X 的最大正整数.一个纠 t 个错误的(n,k)线性码,能够纠正包括 t 个或更少个错误在内的总数为 2^{n-k} 个错误图样.

1.2　标准阵和伴随式译码

设(n,k)线性码 C 的 2^k 个码字为 $v(1) = 0, v(2), \cdots, v(2^k)$.将所有 n 长向量按下述方法分划为 2^k 个互不相交的子集:首先将 C 中的码字排成第一行(全零码字在最左边),从余下的 $2^n - 2^k$ 个 n 长向量中选一个向量 $e(2)$,将 $e(2)$ 与各码字相加后按序排成第二行.再从余下的 n 长向量中选一个向量 $e(3)$,将 $e(3)$ 与各码字相加后按序排成第三行.继续此过程直到用完所有 n 长向量为止.于是就得到如下阵列,称之为码 C 的**标准阵**

$$\begin{matrix}
v(1)=0 & v(2) & \cdots & v(2^k) \\
e(2) & e(2)+v(2) & \cdots & e(2)+v(2^k) \\
\vdots & \vdots & & \vdots \\
e(i) & e(i)+v(2) & \cdots & e(i)+v(2^k) \\
\vdots & \vdots & & \vdots \\
e(2^{n-k}) & e(2^{n-k})+v(2) & \cdots & e(2^{n-k})+v(2^k)
\end{matrix}$$

每个 n 长向量都在标准阵中出现一次,而且仅出现一次.标准阵共有 2^{n-k} 行,且每行含有 2^k 个元素.标准阵的行称为**陪集**,每行中第一个元素 e_i 称为陪集首.线性码能够纠正所有以陪集首为错误图样的错误.各陪集首的伴随式是互不相同的.

设码 C 的一致校验矩阵为 H.接收向量 r 的**伴随式译码算法**可用下面三个步骤完成:

步 1　计算接收向量 r 的伴随式,rH^T;

步 2　确定伴随式等于 rH^T 的陪集首 e_i,然后假定 e_i 是信道引起的错误图样;

步 3　将接收向量 r 译成为码字 $v = r + e_i$.

伴随式译码算法又称为查表法译码,它可以适用于任何线性码.

如果线性码 C 的重量分布为 $A_0,A_1,\cdots,A_n$，它的对偶码的重量分布为 $B_0,B_1,\cdots,B_n$，那么多项式 $A(z)=\sum_{i=0}^{n}A_iz^i$ 和 $B(z)=\sum_{i=0}^{n}B_iz^i$ 之间满足如下恒等式：

$$A(z)=2^{-(n-k)}(1+z)^nB((1-z)/(1+z)).$$

此恒等式称为麦克威廉斯(Mac Williams)恒等式.

1.3　汉　明　码

1. 基本参数

汉明码的一致校验矩阵 H 由所有非零 $n-k$ 长的二元向量为列构成，它的码长 $n=2^m-1$，$m\geqslant 3$ 是正整数，信息位数 $k=2^m-m-1$；一致校验位数 $n-k=m$，纠正错误的能力 $t=1$，最小汉明距离 $d=3$.

2. 重量分布

码长为 $n=2^m-1$ 的汉明码的重量分布 A_i 等于下面多项式 $A(z)$(称为重量枚举式)中 z^i 的系数.

$$A(z)=\{(1+z)^n+n(1-z)(1-z^2)^{(n-1)/2}\}/(1+n).\qquad(1\text{-}2)$$

例3　码长为7的汉明码的重量枚举式为

$$A(z)=\{(1+z)^7+7(1-z)(1-z^2)^3\}/8=1+7z^3+7z^4+z^7,$$

因此其重量分布为 $A_0=1,A_3=A_4=7,A_7=1$，其它各 A_i 为零.

2　非线性码

2.1　码的一些界

一个(n,M,d)(非)线性码其实就是由 M 个长度为 n 的向量(称为码字)组成的如下集合：任意两个不同的码字之间的汉明距离至少为 d. 线性码显然是一个特例. 如果 $2^m<n\leqslant 3\cdot 2^{m-1}$，那么存在$(n,r\cdot 2^{n-m-1},3)$非线性码，其中 $r=20/16$，19/16，或 18/16.

1. 普洛特金(Poltkin)界

如果(n,M,d)码(线性或非线性)满足 $n<2d$，那么必有 $M\leqslant 2[d/(2d-n)]$.

设 $A(n,d)$ 表示所有(n,M,d)码中所含码字个数 M 的最大值，那么

$$A(n,2r-1)=A(n+1,2r),$$
$$A(n,d)\leqslant 2A(n-1,d).$$

若 d 是偶数且 $2d>n$，那么

$$A(n,d)\leqslant 2[d/(2d-n)],\quad A(2d,d)\leqslant 4d;$$

若 d 是奇数且 $2d+1>n$，那么

$$A(n,d) \leqslant 2[(d+1)/(2d+1-n)], \quad A(2d+1,d) \leqslant 4d+4.$$

2. 汉明界

$$A(n,2\delta+1)\left(1+\binom{n}{1}+\cdots+\binom{n}{\delta}\right) \leqslant 2^n.$$

3. Singleton 界

设(n,k)线性码的最小距离是 d,则 $n \geqslant d+k-1$.

4. Griesmer 界

设 $N(k,d)$ 表示满足给定维数 k、最小距离为 d 的最短线性码的码长,则

$$N(k,d) \geqslant \sum_{i=0}^{k-1}\left[\frac{d}{2^i}\right].$$

5. 渐进界

设 $R = \log_2 A(n,d)/n, H_2(x) = -x\log_2 x - (1-x)\log_2(1-x), 0 \leqslant \delta \leqslant 1/2$, 则存在无限多个最小距离为 d 的(n,k)线性码,$d/n \geqslant \delta$, $R = k/n$ 满足:当 $n \to \infty$ 时,$R \geqslant 1 - H_2(d/n)$. 称之为 Gilbert-Varshamov 下界.

6. 球覆盖或汉明上界

对于任意码(n,M,d),当 $n \to \infty$ 时,$R \leqslant 1 - H_2(d/(2n))$.

7. Mc Eliece-Rodemich-Rumsey-Welch 上界

对于任意码(n,M,d),

$$R \leqslant H_2\left(\frac{1}{2} - \left(\frac{d}{n}\left(1-\frac{d}{n}\right)\right)^{1/2}\right).$$

2.2　阿达马码

元素为 $+1$ 或 -1 的 n 阶方阵 H 称为 n 阶阿达马矩阵,当且仅当 $HH^T = nI$. 换句话说,H 是一个正交矩阵. 如果 H 的第一行全为 $+1$,则称它为规一化的阿达马矩阵.

n 阶阿达马矩阵存在的必要条件是 $n = 1, 2$ 或 $4k$.

2^n 阶阿达马矩阵 H_n 可以按如下方法递归地生成:$H_0 = 1$,对非负整数 $n \geqslant 0$,

$$H_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}.$$

将一个规一化的 n 阶阿达马矩阵中的 $+1$ 变为 0,同时将 -1 变为 1,得到矩阵 A 称阿达马码. 若将 A 的第一列去掉,便得到一个$\left(n-1, n, \frac{n}{2}\right)$码 A_1. 若将 A_1 与其互补矩阵重叠,便得到一个$\left(n-1, 2n, \frac{n}{2}-1\right)$码. 若将 A 与其互补矩阵重叠,便得到一个$\left(n, 2n, \frac{n}{2}\right)$码.

只要存在足够的阿达马矩阵,那么普洛特金界是可以达到的.

3　循　环　码

3.1　循环码的描述

(1) 基本定义　将一个 n 长向量 $\boldsymbol{v}=(v_0,v_1,\cdots,v_{n-1})$ 的分量循环右移 i 位 $(1\leqslant i\leqslant n)$，得到 $\boldsymbol{v}^{(i)}=(v_{n-i},v_{n-i+1},\cdots,v_{n-1},v_0,\cdots,v_{n-i-1})$. 把 $\boldsymbol{v}$ 循环右移 i 位等价于将 $\boldsymbol{v}$ 循环左移 $n-i$ 位.

一个 (n,k) 线性码 C，若它的每个码字的每一个循环移位都是 C 中的一个码字，则称此码 C 为一个**循环码**.

例 1　由下列矩阵 $\boldsymbol{H}$ 为一致校验矩阵而得到的线性码是一个(7,4)循环码.

$$\boldsymbol{H}=\begin{bmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}.$$

(2) 多项式描述　每个码字 $\boldsymbol{v}=(v_0,v_1,\cdots,v_{n-1})$ 可唯一地对应于一个码字多项式 $v(x)=\sum_{i=0}^{n-1}v_ix_i$.

一个 (n,k) 循环码 C 中常数项为 1 次数为 $n-k$ 的非零多项式 $g(x)$，($g(x)$ 是唯一存在的) 称为码 C 的**生成多项式**. 每个码字多项式 $v(x)$ 都是生成多项式的倍式，即存在一个消息多项式 $u(x)=\sum_{i=0}^{k-1}u_ix^i$，使得 $v(x)=u(x)g(x)$.

(n,k) 循环码的生成多项式是 $1+x^n$ 的因式. 反过来，若 $g(x)$ 是一个 $n-k$ 次多项式且是 $1+x^n$ 的因式，那么 $g(x)$ 是某个 (n,k) 循环码的生成多项式.

例 2　例 1 所述的(7,4)循环码的生成多项式为 $g(x)=1+x^2+x^3$.

若 (n,k) 循环码 C 的生成多项式为 $g(x)$，那么按以下步骤可将它变成一个系统码.

步 1　用 x^{n-k} 乘以消息多项式 $u(x)$.

步 2　用生成多项式 $g(x)$ 除以 $x^{n-k}u(x)$ 得到余式 $b(x)$(一致校验元).

步 3　联合 $b(x)$ 和 $x^{n-k}u(x)$ 得到码字多项式 $b(x)+x^{n-k}u(x)$.

(3) 矩阵描述　若 (n,k) 循环码 C 的生成多项式为 $g(x)=\sum_{i=0}^{n-k}g_ix^i$，那么以 n 长向量 $\boldsymbol{g}=(g_0,\cdots,g_{n-k},0,\cdots,0)$ 为第一行，以 $\boldsymbol{g}$ 的右移 i 位，$1\leqslant i\leqslant k-1$，作为第 $i+1$ 行所得到的 $k\times n$ 阶矩阵就是循环码 C 的生成矩阵.

称 k 次多项式 $h(x)=(1+x^n)/g(x)$ 为循环码 C 的**校验多项式**. 循环码的对偶码也是循环码，并且其生成多项式为 $x^kh(x^{-1})$.

(4) **循环汉明码**　以 m 次本原多项式为生成多项式得到的循环码是一个长度为 2^m-1 的汉明码.

3.2　循环码的检错能力

设某个循环码 C 的一致校验矩阵为 H，生成多项式为 $g(x)$，那么接收向量 $r=(r_0,\cdots,r_{n-1})$ 的伴随式为 $s=(s_0,s_1,\cdots,s_{n-k-1})=rH^{\mathrm{T}}$，并且多项式 $s(x)=\sum_{i=0}^{n-k-1}s_ix^i$ 等于接收多项式 $r(x)=\sum_{i=0}^{n-1}r_ix^i$ 被 $g(x)$ 除之后所得到的余式，因此也称多项式 $s(x)$ 为 $r(x)$ 的伴随式.

错误位局限于 m 或更少个连续位的错误图样称为长度不超过 m 的突发错误. 如果错误局限于前面 i 位及后面 $m-i$ 位，称此错误图样为长度不超过 m 的首尾相接突发错误.

每个 (n,k) 循环码都能够检测长度不超过 $n-k$ 位的任何突发错误（包括首尾突发错误）. 长度为 $n-k+1$ 位的突发错误不能被检测的概率为 $1/2^{n-k-1}$. 长度为 $m>n-k+1$ 的突发错误不能被检测的概率为 $1/2^{n-k}$.

例 3　由生成多项式 $g(x)=1+x+x^3$ 得到的 $(7,4)$ 循环码的最小距离为 3，它能够检测 2 个随机错误，或检测出长度不超过 3 的突发错误. 它也能检测出很多长度大于 3 的突发错误.

3.3　循环码的编码

(n,k) 循环码的编码可由一个除法电路完成. 该电路是一个有反馈连线的，根据生成多项式 $g(x)=1+\sum_{i=1}^{n-k-1}g_ix^i+x^{n-k}$ 设计的 $(n-k)$ 级线性移存器.

编码运算过程由以下三步完成：

步 1　接通反馈连线并将 k 位信息位移入线路中，同时送入通信信道. 一旦信息全部移入线路，在寄存器中的 $n-k$ 个数据就构成了一致校验数据的余项.

步 2　断开反馈连线.

步 3　移出校验元，并把它们送入信道. 这 $n-k$ 个一致校验元和 k 个信息元一起构成一个完整的码字.

3.4　循环码的译码

设发送的码字多项式是 $v(x)$，信道产生的错误图样是 $e(x)$，于是接收向量多项式为 $r(x)=v(x)+e(x)$.

循环码的译码可分为三个步骤：

步 1　由接收到的 $r(x)$ 计算伴随式多项式 $s(x)$.

步 2　根据伴随式 $s(x)$ 找到错误图样 $e'(x)$.

步3　计算 $r(x)-e'(x)=v'(x)$,得到译码器输出的估计值码字 v'.若 $v'=v$ 则译码正确,否则译码错误.

1. 梅吉特(Meggitt)译码器

梅吉特译码器是(n,k)循环码的通用译码器.它由三个主要部分组成:①伴随式寄存器;② 错误图样检测器;③ 存贮接收向量的缓冲寄存器.

译码算法由以下五个步骤完成:

步1　接收向量全部移入伴随式寄存器,得到伴随式,同时接收向量也存入缓冲寄存器.

步2　把伴随式读入检测器,以检测相应的错误图样.检测器是一个组合逻辑,它输出"1",当且仅当伴随式寄存器中的伴随式与在最高位 x^{n-1} 有错的错误图样相对应.

步3　由缓冲器读出第一个接收符号,与此同时伴随式寄存器移位一次.若检测到第一个接收符号是错误的,则检测器的输出给予纠正,检测器的输出也反馈到伴随式寄存器以修正伴随式.这样便得到一个新的伴随式,它相应于向右移位一次后所得的修正接收向量的伴随式.

步4　由第三步所得到的新伴随式检测第二个接收符号是否有错.译码器重复步2和步3,第二个接收符号的纠正方法和第一个接收符号一样.

步5　如同上述那样,译码器逐个符号地译接收向量,直到从缓冲器中读出全部接收向量为止.

2. 循环汉明码的译码

码长为 2^m-1 循环汉明码的译码电路是一个由其生成多项式确定的移位寄存器.

译码步骤由以下四步完成:

步1　接收向量全部移入伴随式寄存器,得到伴随式,同时接收向量也存入缓冲寄存器.若伴随式为零,则认为没有错误;若伴随式不为零,则译码器认为出现了单个错误.

步2　接收码字一位一位地从缓冲器读出.当每位数据从缓冲器读出时,伴随式寄存器就循环移位一次,一旦寄存器中的伴随式为$(0,0,\cdots,0,1)$,则由缓冲器读出的下一个数据是错误的,因而 m 个输入端的与门输出为"1".

步3　错误数据由缓冲器读出并被有 m 个输入端的与门输出所纠正,用异或门完成该纠错.

步4　接收向量的全部读出缓冲器后,伴随式寄存器重置为零.

3. 捕错译码

捕错译码特别适用于纠突发错误码,纠单个错误码,以及某些低码率和码长较短纠错能力较弱的码.

考虑由 $g(x)$ 生成的(n,k)系统循环码,如果错误位全部集中在校验元的 $n-k$ 位上,那么错误图样就是伴随式.于是,只需将接收向量减去伴随式就行了.此种译码方法称为捕错译码.一般地,如果错误全部集中在 $n-k$ 个连续位上,那么捕错译码都有效.

4 BCH 码

BCH 码是实际中大量使用的一类重要的循环码，是由 Bose R.C. 和 Ray-Chaudhuri D.K.以及 Hocquenghem A.独立发现的.

4.1 BCH码的定义

有限域 GF(2) 上的(n,k)循环码 C 称为一个设计距离为δ 的 **BCH 码**，如果它的生成多项式 $g(x)$ 是 $\alpha^l,\alpha^{l+1},\cdots,\alpha^{l+\delta+1}$ 的极小多项式的最小公倍式，这里 l 是某个整数，α 是一个n 次本原单位根. 若 $l=1$，称之为狭义 BCH 码. 若 $n=2^m-1$，即 α 是 $GF(2^m)$ 的一个本原元，称之为本原 BCH 码.

显然，c 是一个码字，当且仅当 $c(\alpha^l)=c(\alpha^{l+1})=\cdots=c(\alpha^{l+\delta+1})$.

一个设计距离为 δ 的 BCH 码的极小距离至少是 δ，称为 BCH 界.

一般地，找出 BCH 码确切的极小距离是个难题. 但是，对于本原 BCH 码的极小距离有以下熟知的结论：

(1) 码长为 $n=2^m-1$，设计距离为 $\delta=2^t-1$ 的本原 BCH 码的极小距离为 δ.

(2) 一个设计距离为 δ 的本原 BCH 码的极小距离 $d\leqslant 2\delta-1$.

(3) 本原 BCH 码的极小距离为奇数.

4.2 BCH码的译码算法

设 C 是码长为n、设计距离为 $\delta=2t+1$ 的 BCH码，α 是有限域 $GF(2^m)$ 的一个 n 次本原单位根. 假设传送的码字是 $\boldsymbol{c}=(c_0,c_1,\cdots,c_{n-1})$. 接受到的向量是 $\boldsymbol{y}=\boldsymbol{c}+\boldsymbol{e}$，$\boldsymbol{e}=(e_0,e_1,\cdots,e_{n-1})$ 是差错向量. BCH 码的译码算法可以分以下三步完成.

步 1 计算伴随式.

不妨假设 C 的奇偶校验矩阵为

$$\boldsymbol{H}=\begin{bmatrix}1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1}\\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & \alpha^{\delta-2} & \alpha^{2(\delta-2)} & \cdots & \alpha^{(\delta-2)(n-1)}\end{bmatrix}.$$

令 $c(x)=\sum c_ix^i,e(x)=\sum e_ix^i,y(x)=\sum y_ix^i$，则向量 $\boldsymbol{y}$ 的伴随式为

$$\boldsymbol{S}=\boldsymbol{H}\boldsymbol{y}^{\mathrm{T}}=\begin{bmatrix}1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1}\\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & \alpha^{\delta-2} & \alpha^{2(\delta-2)} & \cdots & \alpha^{(\delta-2)(n-1)}\end{bmatrix}\begin{bmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\end{bmatrix}$$

$$= \begin{bmatrix} \sum y_i\alpha^i \\ \sum y_i\alpha^{3i} \\ \vdots \\ \sum y_i\alpha^{(\delta-2)i} \end{bmatrix} = \begin{bmatrix} y(\alpha) \\ y(\alpha^3) \\ \vdots \\ y(\alpha^{\delta-2}) \end{bmatrix} = \begin{bmatrix} A_1 \\ A_3 \\ \vdots \\ A_{\delta-2} \end{bmatrix},$$

其中 $A_l = y(\alpha^l)$. 假设用 α^l 的极小多项式 $M^{(l)}(x)$ 除 $y(x)$，即

$$y(x) = Q(x)M^{(l)}(x) + R(x), \quad \deg R(x) < \deg M^{(l)}(x),$$

那么 $A_l = y(\alpha^l)$ 恰好就是 $R(x)$ 在 $x = \alpha^l$ 的取值.

类似地，如果需要也可以容易地计算出 $A_2 = A_1^2, A_4 = A_2^2, A_{\delta-1} = A_{(\delta-1)/2}^2$.

步 2　寻找差错位多项式 $\sigma(z)$. 假设向量 e 的重量是 w，非零分量为 $e_{i_1}, \cdots, e_{i_w}$，则 $i_1, \cdots, i_w$ 就是向量 y 中分量出现差错的坐标. 定义 $X_r = \alpha^{i_r}, r = 1, 2, \cdots, w$，为差错位子，差错位多项式定义为

$$\sigma(z) = \prod_{i=1}^{w}(1 - X_i z) = \sum_{i=0}^{w}\sigma_i z^i. \tag{4-1}$$

根据设计距离为 δ 的 BCH 码的定义，$c(\alpha^l) = 0, 1 \leqslant l \leqslant \delta - 1$，所以

$$A_l = y(\alpha^l) = c(\alpha^l) + e(\alpha^l)$$
$$= e(\alpha^l) = \sum_{i=1}^{w} X_i^l.$$

这一步所剩下的任务就是从第一步的 $A_1, A_2, \cdots, A_{\delta-1}$ 中确定差错位多项式 $\sigma(z)$. 要注意的是并不能确定唯一的差错向量 e 或 $\sigma(z)$. 译码器必须寻找满足牛顿恒等式的具有最小重量的差错向量 e.

方法一　Berlekamp 算法. 利用广义牛顿恒等式

$$A_{j+w} + \sigma_1 A_{j+w-1} + \cdots + \sigma_w A_j = 0, \quad \text{对任意 } j. \tag{4-2}$$

可以把 A_l 看作是 w 阶线性反馈移位寄存器初始状态为 $A_1, A_2, \cdots, A_w$ 时的输出. 因此，译码器的任务就是寻找最短长度 w 的线性反馈移位寄存器，使得初始状态为 $A_1, A_2, \cdots, A_w$ 时，输出为 $A_1, A_2, \cdots, A_{\delta-1}$. 对此 Berlekamp 已经给出了有效算法.

方法二　利用牛顿恒等式. 如果发生 w 个差错，则

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ A_2 & A_1 & 1 & 0 & 0 & \cdots & 0 \\ A_4 & A_3 & A_2 & A_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{2w-4} & A_{2w-5} & \cdots & \cdots & \cdots & \cdots & w-3 \\ A_{2w-2} & A_{2w-3} & \cdots & \cdots & \cdots & \cdots & w-1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{w-1} \\ \sigma_w \end{bmatrix} = \begin{bmatrix} A_1 \\ A_3 \\ A_5 \\ \vdots \\ A_{2w-3} \\ A_{2w-1} \end{bmatrix}. \tag{4-3}$$

假设发生 $w \leqslant t$ 个差错，对于设计距离为 $\delta = 2t + 1$ 的 BCH 码来说，有一个递归算法计算 $\sigma(z)$. 若有 t 或 $t - 1$ 个差错，在方程(4-3) 中用 t 代替 w，方程(4-3) 有解；若差错个数少于 $t - 1$，方程(4-3) 无解，这时假设有 $t - 2$ 个差错，用 $t - 2$ 代替

方程(4-3)中的 w,重复上述过程直到找到一个解.

步 3 计算差错位多项式 $\sigma(z)$ 的根.由于 $\sigma(z)=\prod_{i=1}^{w}(1-X_i z)$,所以 $\sigma(z)$ 的零点的倒数是 $X_1=\alpha^{i_1},\cdots,X_w=\alpha^{i_w}$,发生差错的位是 $i_1,i_2,\cdots,i_w$.最简单的方法就是验证所有的 α 的方幂,$\sigma(\alpha^{-i})=0$ 当且仅当在位置 i 上有差错.

5 MDS 码

MDS码,即最大距离可分码,是代数编码理论中最吸引人的部分之一.对于任意(n,k)线性码 C,都有 Singleton 界:最小距离 $d\leqslant n-k+1$.极端情况 $d=n-k+1$ 时,称 C 为 **MDS码**.之所以称之为MDS码,是因为①码字之间具有最大可能距离,②每个码字可以分成信息位和校验位两部分,也就是说,这种码有系统编码器.

5.1 MDS码的奇偶校验矩阵和生成矩阵

设 C 是一个(n,k)线性码,最小距离为 d,奇偶校验矩阵和生成矩阵分别为 $\boldsymbol{H}$ 和 $\boldsymbol{G}$.以下说法是等价的:

(1) C 是 MDS码.

(2) 生成矩阵 $\boldsymbol{G}$ 的任意 k 列是线性独立的.

(3) 奇偶校验矩阵 $\boldsymbol{H}$ 的任意 $n-k$ 列是线性独立的.

根据奇偶校验矩阵和生成矩阵的关系,容易得到 MDS 码的对偶码也是 MDS码.

假设一个(n,k)线性码 C 的最小距离为d,生成矩阵为 $\boldsymbol{G}=[\boldsymbol{I}\mid\boldsymbol{A}]$,$\boldsymbol{A}$ 是一个 $k\times(n-k)$ 矩阵,则 C 是 MDS码,当且仅当 $\boldsymbol{A}$ 的每个子方阵是非奇异的.

5.2 MDS码的重量分布

设 C 是有限域 GF(q) 上的一个(n,k)MDS码,$d=n-k+1$,A_w 表示重量为w 的码字个数,那么

$$A_w=\binom{n}{w}(q-1)\sum_{i=0}^{w-d}(-1)^i\binom{w-1}{i}q^{w-d-i}.$$

5.3 MDS码与有限射影几何

设射影几何 PG$(k-1,q)$ 上 n 个点的集合 S 具有这样的性质:S 中任意 k 个点都是线性独立的,即不存在 k 个点恰好在同一个超平面上.称这样的集合 S 为一个 n 弧.

例1　设 $q=2^m,\alpha_1,\alpha_2,\cdots,\alpha_{q-1}$ 是有限域 GF(q) 的所有非零元素. 矩阵

$$\begin{bmatrix} 1 & \cdots & 1 & 1 & 0 & 0 \\ \alpha_1 & \cdots & \alpha_{q-1} & 0 & 1 & 0 \\ \alpha_1^2 & \cdots & \alpha_{q-1}^2 & 0 & 1 & 0 \end{bmatrix}$$

有 2^m+2 个列向量,并且任意3个列向量都是线性独立的,所以列向量的集合是射影几何 PG$(2,2^m)$ 上的一个(2^m+2) 弧.

设 q 是奇数,对于射影几何 PG$(k-1,q)$ 上的任意 n 弧来说,$n\leqslant q+k-2$.

根据上节的 MDS 码的等价条件,构造一个(n,k)MDS 码的问题等价于构造射影几何 PG$(k-1,q)$ 上的 n 弧问题.

给定 k 和 q,计算最大值 n 使得在有限域 GF(q) 上存在(n,k)MDS 码是一个很有兴趣但很困难的问题. 令 $m(k,q)$ 表示 n 的最大值.

MDS 码主要猜想:除了 $q=2^m,m(3,q)=m(q-1,q)=q+2$ 之外,

$$m(k,q)=\begin{cases} q+1, & 2\leqslant k\leqslant q, \\ k+1, & q<k. \end{cases}$$

这个猜想对于 $k\leqslant 5,q\leqslant 11,q>(4k-9)^2$ 已经证明是成立的.

6　R-S 码

R-S(Reed-Solomon) 码是一类特殊的 BCH 码,它具有很大的理论和实用价值. 主要体现在:

(1) 要求码长小于有限域的元素个数时,R-S码是理想的. 并且作为 MDS 码,它具有最大可能的最小距离.

(2) 从 R-S 码出发可以构造出多种类型的好码.

(3) 利用 R-S 码可以纠突发差错.

6.1　R-S 码的定义

有限域 GF(q) 上码长为 $n=q-1$ 的 BCH 码称为 **R-S 码**,当然 $q\neq 2$. R-S码是循环码,设计距离为 δ 的 R-S 码的生成多项式为

$$g(x)=(x-\alpha^l)(x-\alpha^{l+1})\cdots(x-\alpha^{l+\delta-2}),$$

其中 α 是 GF(q) 的本原元.

R-S 码的对偶也是 R-S 码.

R-S 码的维数 $k=n-\delta+1$,根据 BCH 界和 Singleton 界,容易得到最小距离 $d=n-k+1$. 换句话说,R-S 码是 MDS 码. 因而其重量分布由 5.2 节给出.

设 $n=q-1,\boldsymbol{\beta}=(\beta_1,\beta_2,\cdots,\beta_n),\beta_i\neq\beta_j,i\neq j,\beta_i\in$ GF(q), $\boldsymbol{v}=(v_1,v_2,\cdots,v_n),v_i\in$ GF$(q)^*$,称向量的集合 GRS$_k(\boldsymbol{\beta},\boldsymbol{v})$ 为**广义 R-S 码**,其中

GRS$_k(\boldsymbol{\beta},\boldsymbol{v})=\{(v_1F(\beta_1),v_2F(\beta_2),\cdots,v_nF(\beta_n))\mid F(z)$ 是 GF(q) 上次数

小于 k 的多项式}.

当所有的 $v_i = 1$ 时, $GRS_k(\boldsymbol{\beta}, \boldsymbol{v})$ 就是一个 R-S 码.

由于 $F(z)$ 至多有 $k-1$ 个零点,所以广义 R-S码的最小距离至少是 $n-k+1$,根据 Singleton 界应该等于 $n-k+1$,即广义 R-S 码是 MDS 码.

存在向量 $\boldsymbol{y}$ 使得广义 R-S 码 $GRS_k(\boldsymbol{\beta}, \boldsymbol{v})$ 的对偶码是一个广义 R-S 码 $GRS_{n-k}(\boldsymbol{\beta}, \boldsymbol{y})$,其生成矩阵是

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_n \\ \vdots & \vdots & & \vdots \\ \beta_1^{n-k-1} & \beta_2^{n-k-1} & \cdots & \beta_n^{n-k-1} \end{bmatrix} \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & y_n \end{bmatrix} \tag{6-1}$$

6.2 R-S 码的编码

方法一 由于 R-S 码是循环码,当然可以利用循环码的编码器.

方法二 设 $\boldsymbol{u} = (u_0, u_1, \cdots, u_{k-1}), u_i \in GF(q)$ 是被编码信息, $u(z) = \sum_{i=0}^{k-1} u_i z^i$,则信息 $\boldsymbol{u}$ 对应的码字为

$$\boldsymbol{c} = (u(1), u(\alpha), \cdots, u(\alpha^{n-1})). \tag{6-2}$$

所有向量 $\boldsymbol{c}$ 组成的集合恰好就是一个 R-S 码.这个编码方法不是系统的.

6.3 R-S 码的译码

方法一 由于 R-S码是 BCH码,所以 BCH码的译码算法对 R-S码也是适用的.

方法二 大数逻辑译码.假设信道传输的码字如(6-2)式,差错向量是 $\boldsymbol{e} = (e_0, e_1, \cdots, e_{n-1})$,接收到 $\boldsymbol{y} = (y_0, y_1, \cdots, y_{n-1})$.那么译码器知道以下 n 个方程组

$$\begin{aligned} y_0 &= e_0 + u_0 + u_1 + u_2 + \cdots + u_{k-1}, \\ y_1 &= e_1 + u_0 + \alpha u_1 + \alpha^2 u_2 + \cdots + \alpha^{k-1} u_{k-1}, \\ &\cdots\cdots \\ y_{n-1} &= e_{n-1} + u_0 + \alpha^{n-1} u_1 + \alpha^{2(n-1)} u_2 + \cdots + \alpha^{(k-1)(n-1)} u_{k-1}. \end{aligned}$$

译码器从以上的 n 个方程组中任取 k 个都可以确定唯一的向量 $\boldsymbol{u}' = (u'_0, u'_1, \cdots, u'_k)$,共有 $\binom{n}{k}$ 种取法,因而得到 $\binom{n}{k}$ 个向量(有相同的).假设 $\boldsymbol{e}$ 的重量是 w,即发生 w 个差错,那么信息 $\boldsymbol{u}$ 至少出现 $\binom{n-w}{k}$ 次,其它向量至多出现 $\binom{w+k-1}{k}$ 次.

所以当 R-S码的最小距离 $d > 2w$,即 $\binom{n-w}{k} > \binom{w+k-1}{k}$ 时,译码器能正确地

译出信息.

6.4 R-S码的应用

1. R-S码的像

设 C 是有限域 $GF(2^m)$ 上的一个(n,k)R-S码,最小距离是 d.假设 $\beta_1,\beta_2,\cdots,\beta_m$ 是 $GF(2^m)$ 在 $GF(2)$ 上的一个基,则码 C 的任意码字 c 的分量 c_i 可以表示为 $c_i=\sum_{j=1}^{m}c_{ij}\beta^j$.即把码字 c 映射到$(c_{11},c_{12},\cdots,c_{1m},\cdots,c_{n1},c_{n2},\cdots,c_{nm})$.这样码 C 的像为 $GF(2)$ 上的(nm,mk)线性码,其最小距离 $d'\leqslant d$.一般可以增大码的最小距离.

2. R-S码的纠突发差错能力

实际上,许多信道中出现差错都是突发的,而不是随机的.对于这样的信道,使用R-S码特别合适.假设二元信息取成一列 m 个符号,看作有限域 $GF(2^m)$ 中的元素.由于含有 b 个差错的一个突发错误仅仅影响 $GF(2^m)$ 中 r 个相邻符号,其中 $(r-2)m+2\leqslant b\leqslant(r-1)m+1$,所以,如果用于编码的R-S码的最小距离 d 比 r 大许多,则可以纠多个突发错误.

3. R-S码的扩展

给一个码添加一致奇偶校验并不一定能增大最小距离.但对于R-S码来说,添加一致奇偶校验一定能增大最小距离.设 C 是有限域 $GF(q)$ 上的一个(n,k)R-S码,最小距离为 d,生成多项式为 $g(x)=(x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{d-1})$,$\alpha$ 是 $GF(q)$ 上的本原元.那么通过添加一致奇偶校验得到的扩展R-S码的最小距离是 $d+1$.

4. 构造 Justesen 码

我们知道长BCH码是坏码.按照本节1中介绍的方法,从R-S码得到的长2元码也是坏码.然而,从R-S码构造的Justesen码是好码.

假设 C 是有限域 $GF(2^m)$ 上的$(2^m-1,k)$R-S码,当然其最小距离是 $d=2^m-k$.α 是 $GF(2^m)$ 的一个本原元.令 $\boldsymbol{a}=(a_0,a_1,\cdots,a_{n-1})$,$a_i\in GF(2^m)$,是 C 的任意码字.令向量 $\boldsymbol{b}(\boldsymbol{a})=(a_0,a_0;a_1,\alpha a_1;a_2,\alpha^2a_2;\cdots;a_{n-1},\alpha^{n-1}a_{n-1})$.类似本节1的方法,向量 $\boldsymbol{b}(\boldsymbol{a})$ 对应一个长度为$2mn$的向量,记为 $\boldsymbol{b}(\boldsymbol{a})$.称集合 $J_{n,k}=\{\boldsymbol{b}(\boldsymbol{a})\mid\boldsymbol{a}\in C\}$ 为Justesen码.它是一类级联码.

5. 构造 MDS 码

设有限域 $GF(q)$ 上的所有非零元素为 $\alpha_1,\alpha_2,\cdots,\alpha_{q-1}$.由本节3得到的$(q,k)$扩展R-S码的奇偶校验矩阵为

$$\begin{bmatrix}1&\cdots&1&1\\ \alpha_1&\cdots&\alpha_{q-1}&0\\ \alpha_1^2&\cdots&\alpha_{q-1}^2&0\\ \vdots&&\vdots&\vdots\\ \alpha_1^{q-k-1}&\cdots&\alpha_{q-1}^{q-k-1}&0\end{bmatrix}.$$

在此基础上再添加一些一致奇偶校验就可以构造一个$(q+1,k)$MDS码,其奇偶校验矩阵是

$$\begin{bmatrix} 1 & \cdots & 1 & 1 & 0 \\ \alpha_1 & \cdots & \alpha_{q-1} & 0 & 0 \\ \alpha_1^2 & \cdots & \alpha_{q-1}^2 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots \\ \alpha_1^{q-k-1} & \cdots & \alpha_{q-1}^{q-k-1} & 0 & 0 \\ \alpha_1^{q-k} & \cdots & \alpha_{q-1}^{q-k} & 0 & 1 \end{bmatrix}.$$

7 交 错 码

交错码是通过稍微修正BCH码的奇偶校验矩阵得到的一大类非常重要的码,其中一些长码可以达到Gilbert-Varshamov界而且还包含一些性质很好的子码.

7.1 交错码的定义

根据6.1节,存在向量$\boldsymbol{y}$使得广义R-S码$\mathrm{GRS}_k(\boldsymbol{\beta},\boldsymbol{v})$的对偶码是一个广义R-S码$\mathrm{GRS}_{n-k}(\boldsymbol{\beta},\boldsymbol{y})$,其生成矩阵如(6-1)式.定义集合$\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})=\{\boldsymbol{a}\in\mathrm{GRS}_k(\boldsymbol{\beta},\boldsymbol{v})\mid a_i\in\mathrm{GF}(q),i=1,2,\cdots,n\}$为**交错码**.实际上,交错码$\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$就是广义R-S码$\mathrm{GRS}_k(\boldsymbol{\beta},\boldsymbol{v})$在有限域$\mathrm{GF}(q)$上的限制,或者说就是$\mathrm{GF}(q)$上满足$\boldsymbol{Ha}^{\mathrm{T}}=\boldsymbol{0}$的所有向量$\boldsymbol{a}$的集合.令$r=n-k$,那么交错码$\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$的维数$k$满足$n-mr\leqslant k\leqslant n-r$,最小距离$d\geqslant r+1$.

设有限域$\mathrm{GF}(q^m)$到$\mathrm{GF}(q)$上的迹函数为T_m,C是$\mathrm{GF}(q^m)$上的任意一个码,那么$\mathrm{GF}(q)$上的**迹码**定义为

$$T_m(C)=\{(T_m(c_1),T_m(c_2),\cdots,T_m(c_n))\mid(c_1,c_2,\cdots,c_n)\in C\}.$$

交错码$\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$的对偶码是$T_m(\mathrm{GRS}_{n-k}(\boldsymbol{\beta},\boldsymbol{y}))$.

7.2 交错码的译码

交错码的译码算法类似BCH码的译码.假设$\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$的最小距离大于等于$r+1$,$r$是偶数.差错向量$\boldsymbol{e}$的重量为$t\leqslant r/2$,其差错位是$X_1=\alpha^{i_1},X_2=\alpha^{i_2},\cdots,X_t=\alpha^{i_t}$,差错值是$Y_1=e_{i_1},Y_2=e_{i_2},\cdots,Y_t=e_{i_t}$.

步1 计算伴随式$\boldsymbol{S}=\boldsymbol{He}^{\mathrm{T}}=(s_0,s_1,\cdots,s_{r-1})^{\mathrm{T}}$,其中$s_u=\sum_{v=1}^{t}X_v^uY_vy_{i_v}$,$u=0,1,\cdots,r-1$.

步2 利用多项式的欧几里德算法计算差错位和差错赋值多项式

$$w(z)=\sum_{v=1}^{t}Y_vy_{i_v}\prod_{u=1,u\neq v}^{t}(1-X_uz).$$

可以证明

$$w(z)/\sigma(z) \equiv S(z)\ (\bmod z^r), \tag{7-1}$$

其中 $S(z) = \sum_{u=0}^{r-1} s_u z^u$，$\sigma(z)$ 是差错位多项式. 称(7-1)式为**关键方程**. 这步的目的是，对于给定的 $S(z)$，寻找 $\sigma(z)$ 和 $w(z)$ 使(7-1)式成立，并且 $\sigma(z)$ 的次数尽可能小. 利用多项式的欧几里德算法可以做到.

步3　差错位 X_i 就是 $\sigma(z)$ 的根的倒数，差错值为

$$Y_u = \frac{w(X_u^{-1})}{y_{i_u}\prod_{v\neq u}(1 - X_v X_u^{-1})}.$$

7.3　Goppa码

Goppa码是交错码中最令人感兴趣的一类子码，长的Goppa码是优码，而且存在一系列Goppa码满足Gilbert-Varshamov界.

设 $G(z)$ 是 $\mathrm{GF}(q^m)$ 上的多项式，$\mathrm{GF}(q^m)$ 的子集 $L = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ 满足 $G(\alpha_i) \neq 0, i = 1,2,\cdots,n$. $\mathrm{GF}(q)$ 上所有满足 $\sum_{i=1}^{n}\frac{a_i}{z-\alpha_i} \equiv 0\ (\bmod G(z))$ 的向量 $\boldsymbol{a}$ 的集合 $\Gamma(L,G)$ 称为**Goppa码**. 等价地，使多项式环 $\mathrm{GF}(q^m)[z]/G(z)$ 中 $\sum_{i=1}^{n}\frac{a_i}{z-\alpha_i} = 0$ 成立的 $\mathrm{GF}(q)$ 上所有向量的集合就是Goppa码. 称多项式 $G(z)$ 为Goppa码 $\Gamma(L,G)$ 的Goppa多项式. 如果 $G(z)$ 是不可约的，称 Γ 为**不可约Goppa码**；如果 $G(z)$ 没有重根，称 Γ 为**可分Goppa码**.

显然，Goppa多项式在Goppa码中扮演的角色类似于生成多项式在循环码中扮演的角色.

假设Goppa多项式 $G(z) = \sum_{i=0}^{r} g_i z^i$，$g_i \in \mathrm{GF}(q^m)$，$g_r \neq 0$，那么Goppa码 $\Gamma(L,G)$ 的奇偶校验矩阵为

$$\begin{bmatrix} g_r & 0 & 0 & \cdots & 0 \\ g_{r-1} & g_r & 0 & \cdots & 0 \\ g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_r \end{bmatrix}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \cdots & \alpha_n^{r-1} \end{bmatrix}\times$$

$$\begin{bmatrix} G(\alpha_1)^{-1} & & & 0 \\ 0 & G(\alpha_2)^{-1} & & 0 \\ & & \ddots & \\ 0 & & & G(\alpha_n)^{-1} \end{bmatrix}, \tag{7-2}$$

因而，Goppa码是交错码 $\mathscr{A}(\boldsymbol{\alpha}, \boldsymbol{y})$，其中

$$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n), \boldsymbol{y} = (G(\alpha_1)^{-1}, \cdots, G(\alpha_n)^{-1}),$$

所以 Goppa 码的维数 $k \geqslant n - mr$，最小距离 $d \geqslant r + 1$，$r = \deg G(z)$.

当 $G(z) = z^{2t}$，$L = GF(2^m)^*$ 时，Goppa 码就是纠 t 个错的 BCH 码.

Goppa 码 $\Gamma(L,G)$ 是广义 R-S 码 $GRS_{n-r}(\boldsymbol{\alpha},\boldsymbol{v})$ 在 $GF(q)$ 上的限制，

$$\boldsymbol{v} = (v_1, v_2, \cdots, v_n),\ v_i = G(\alpha_i)/\prod_{j\neq i}(\alpha_i - \alpha_j),\ i = 1,\cdots,n.\ \Gamma(L,G)$$

的对偶码是 $T_m(GRS_r(\boldsymbol{\alpha},\boldsymbol{y}))$.

7.4 广义 Srivastava 码

由交错码 $\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$ 的定义知道，假设 $[c_{ij}]$，$c_{ij} \in GF(q^m)$ 是任意可逆矩阵，那么

$$H' = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1r} \\ c_{21} & c_{22} & \cdots & c_{2r} \\ \vdots & \vdots & & \vdots \\ c_{r1} & c_{r2} & \cdots & c_{rr} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_n \\ \vdots & \vdots & & \vdots \\ \beta_1^{r-1} & \beta_2^{r-1} & \cdots & \beta_n^{r-1} \end{bmatrix} \cdot \begin{bmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_n \end{bmatrix}$$

$$= \begin{bmatrix} y_1 g_1(\beta_1) & y_2 g_1(\beta_2) & \cdots & y_n g_1(\beta_n) \\ y_1 g_2(\beta_1) & y_2 g_2(\beta_2) & \cdots & y_n g_2(\beta_n) \\ \vdots & \vdots & & \vdots \\ y_1 g_r(\beta_1) & y_2 g_r(\beta_2) & \cdots & y_n g_r(\beta_n) \end{bmatrix} \tag{7-3}$$

也是一个奇偶校验矩阵，其中 $g_i(x) = c_{i1} + c_{i2}x + c_{i3}x^2 + \cdots + c_{ir}x^{r-1}$，$i = 1,2,\cdots,r$.

在交错码 $\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$ 的奇偶校验矩阵(7-3)式中，令 $r = st$，$\beta_1,\beta_2,\cdots,\beta_n,w_1,w_2,\cdots,w_s$ 是有限域 $GF(q^m)$ 中 $n + s$ 个不同的元素，$z_1,z_2,\cdots,z_n$ 是 $GF(q^m)$ 中非零元素. 令

$$g_{(l-1)t+k}(x) = \prod_{i=1}^{s}(x - w_j)^t/(x - w_l)^k,\ l = 1,\cdots,s,\ k = 1,\cdots,t,$$

$$y_i = z_i/\prod_{j=1}^{s}(\beta_i - w_j)^t$$

且满足

$$y_i g_{(l-1)t+k}(\beta_i) = z_i/(\beta_i - w_l)^k,\ i = 1,2,\cdots,n.$$

这样矩阵(7-3)变为

$$H' = [H_1, H_2, \cdots, H_s]^{\mathrm{T}}，其中$$

$$H_l = \begin{bmatrix} z_1/(\beta_1 - w_l) & z_2/(\beta_2 - w_l) & \cdots & z_n/(\beta_n - w_l) \\ z_1/(\beta_1 - w_l)^2 & z_2/(\beta_2 - w_l)^2 & \cdots & z_n/(\beta_n - w_l)^2 \\ \vdots & \vdots & & \vdots \\ z_1/(\beta_1 - w_l)^t & z_2/(\beta_2 - w_l)^t & \cdots & z_n/(\beta_n - w_l)^t \end{bmatrix}.$$

以此矩阵为奇偶校验矩阵的交错码称为**广义 Srivastava 码**.

特别地，当 $t = 1$，$z_i = \beta_i^{\mu}$ 时，就是 Srivastava 码，它是 Goppa 码.

当 $m = 1$ 时，广义 Srivastava 码是 MDS 码，有时称之为 Gabidulin 码.

7.5 Chien-Choy广义 BCH 码

假设 $\gcd(n,q)=1$,$GF(q^m)$ 是包含所有 n 次单位根的 $GF(q)$ 的最小扩域.符号$[f(z)]_n$ 表示多项式$f(z)$对于除式 z^n-1 的剩余多项式.令 $P(z)$ 和 $G(z)$ 是多项式环 $GF(q^m)[z]$ 中与 z^n-1 互素的多项式,并且 $\deg P(z)\leqslant n-1$,$r=\deg G(z)\leqslant n-1$.那么 Chien-Choy 广义 BCH 码定义为

$$\mathrm{GBCH}(P,G)=\{(a_0,\cdots,a_{n-1})\in GF(q)^n\mid[A(z)P(z)]_n\equiv 0\ (\mathrm{mod}\,G(z))\},$$

其中 $A(z)=\sum_{j=1}^{n}A_jz^{n-j}$,$A_j=a(\alpha^j)$是向量 $\boldsymbol{a}=(a_0,a_1,\cdots,a_{n-1})$的 Mattson-Solomon 多项式,$a(z)=\sum_{i=0}^{n-1}a_iz^i$.

设 $P(z)=\sum_{i=0}^{n-1}p_iz^i$,$G(z)=\sum_{i=0}^{n-1}g_iz^i$,那么 Chien-Choy 广义 BCH 码 GBCH(P,G)的奇偶校验矩阵是

$$\begin{bmatrix}1&1&1&\cdots&1\\1&\alpha&\alpha^2&\cdots&\alpha^{n-1}\\\vdots&\vdots&\vdots&\vdots&\vdots\\1&\alpha^{r-1}&\alpha^{2(r-1)}&\cdots&\alpha^{(n-1)(r-1)}\end{bmatrix}\times\begin{bmatrix}p_0/g_0&&&&0\\&p_1\alpha/g_1&&&0\\&&p_2\alpha^2/g_2&&0\\&&&\ddots&\\0&&&&p_{n-1}\alpha^{n-1}/g_{n-1}\end{bmatrix}.\qquad(7\text{-}4)$$

所以,GBCH(P,G) 是交错码 $\mathscr{A}(\boldsymbol{\beta},\boldsymbol{y})$,这里所有 β_i 的集合是所有 n 次单位根的集合,$y_i=p_{i-1}\alpha^{i-1}/g_{i-1}$.维数 $k\geqslant n-rm$,最小距离 $d\geqslant r+1$.

当 $P(z)=z^{\delta+i-2}$,$G(z)=z^{\delta-1}$ 时,GBCH(P,G) 就是 BCH 码.当 $q=2$ 时,GBCH(z^{n-1},G) 是 Goppa 码 $\Gamma(L,G)$,L 为所有 n 次单位根的集合.

Chien-Choy 广义 BCH 码还有以下的等价定义:

$\boldsymbol{a}\in$ GBCH(P,G) 当且仅当 $\deg[A(z)P(z)/G(z)]_n\leqslant n-1-r$.

GBCH(P,G) = GBCH(P^*,z^r),其中 $P(z)^*=[z^rP(z)/G(z)]_n$.

8 R-M 码

Reed-Muller(R-M) 码的历史相对较长,对它理解的也最深刻.R-M 码与几何码有着密切的关系.采用大数逻辑译码方法非常实用.然而,人们对高阶 R-M 码的认识还不是很多.

8.1 r阶R-M码

1. 定义

假设$f(v_1,v_2,\cdots,v_m)$是$GF(2)^m$上的m元布尔函数.那么布尔函数f可以由1，$v_1,v_2,\cdots,v_m,v_1v_2,v_1v_3,\cdots,v_{m-1}v_m,\cdots,v_1v_2\cdots v_m$的线性和来表示.若$\alpha$是$GF(2^m)$的本原元,称$\boldsymbol{f}=(f(0),f(1),f(\alpha),\cdots,f(\alpha^i),\cdots,f(\alpha^{2m-2}))$为布尔函数$f$对应的向量.

称$GF(2)^m$上所有次数不超过r的布尔函数f对应的向量$\boldsymbol{f}$的集合$\mathscr{R}(r,m)$为r阶R-M码,$0\leqslant r\leqslant m$.

r阶R-M码$\mathscr{R}(r,m)$的码长是2^m,维数是$k=1+\binom{m}{1}+\binom{m}{2}+\cdots+\binom{m}{r}$,最小距离$d=2^{m-r}$.

任意码字的重量都是$2^{\lceil m/r\rceil-1}$的倍数.

$\mathscr{R}(r,m)$和$\mathscr{R}(m-r-1,m)$互为对偶码,$0\leqslant r\leqslant m-1$.

2. R-M码与有限几何的联系

设S是欧几里德几何$EG(m,2)$中的点集,称2^m维向量$\chi(S)$为S的关联向量,如果其第s个分量$\chi(S)_s=\begin{cases}1,s\in S,\\0,s\in S.\end{cases}$

r阶R-M码中最小重量码字恰好是欧几里德几何$EG(m,2)$上$m-r$维平面的关联向量,因而$A_{2^{m-r}}=2^r\prod\limits_{i=0}^{m-r-1}\dfrac{2^{m-i}-1}{2^{m-r-i}-1}$,并且所有最小重量码字生成$\mathscr{R}(r,m)$.

$\mathscr{R}(r,m)$是扩展的循环码,它的自同构群是一般仿射群$GA(m)$,$|GA(m)|=2^m\prod\limits_{i=0}^{m-1}(2^m-2^i)$.

8.2 一阶R-M码

1. 基本性质

一阶R-M码的码率很低,但能纠许多差错,因而非常适用于干扰强烈的信道.

一阶R-M码$\mathscr{R}(1,m)$由所有线性布尔函数对应的向量组成,其重量分布是$A_0=A_{2^m}=1,A_{2^{m-1}}=2^{m+1}-2$.

设$\boldsymbol{u}=(u_1,u_2,\cdots,u_m)\in GF(2)^m$,$f$是任意$m$元布尔函数,对应的向量是$\boldsymbol{f}$.用$f(\boldsymbol{u})$表示向量$\boldsymbol{f}$的第$u$个分量.定义$F(\boldsymbol{u})=(-1)^{f(\boldsymbol{u})}$,那么$\boldsymbol{F}=(F(\boldsymbol{u}))_{\boldsymbol{u}\in GF(2)^m}$是$-1$、$1$向量.称$\hat{F}(\boldsymbol{u})=\sum\limits_{v\in GF(2)^m}(-1)^{\boldsymbol{u}\cdot\boldsymbol{v}+f(\boldsymbol{v})}$为向量$\boldsymbol{F}$的阿达马变换,$\hat{\boldsymbol{F}}$是$2^m$维向量.

包含向量$\boldsymbol{f}$的$\mathscr{R}(1,m)$的陪集的重量分布是$\frac{1}{2}\{2^m\pm\hat{F}(\boldsymbol{u})\}$,$\boldsymbol{u}\in GF(2)^m$,它

即是向量 f 与 $\mathscr{R}(1,m)$ 中码字的距离分布.

2. 译码算法

根据极大似然译码算法,若收到的向量是 f,译码器必须计算向量 f 与 $\mathscr{R}(1,m)$ 中每个码字的距离,然后把 f 译为与其距离最小的码字.

根据上节中向量 f 与 $\mathscr{R}(1,m)$ 中码字的距离分布,译码器必须找到最大的 $|\hat{F}(u)|$. 如果 $\hat{F}(u)\geqslant 0$,则把 f 译为布尔函数 $\sum_{i=1}^{m}u_iv_i$ 对应的码字;如果 $\hat{F}(u)<0$,则把 f 译为布尔函数 $1+\sum_{i=1}^{m}u_iv_i$ 对应的码字.

8.3 二阶 R-M 码

1. 与一阶 R-M 码的关系

称主对角线元素为零的二元对称方阵为辛矩阵. 设 $B=[B_{ij}]$ 是一个辛矩阵, 称布尔函数 $\mathscr{B}(u_1,\cdots,u_m,v_1,\cdots,v_m)=\sum_{i,j=1}^{m}u_iB_{ij}v_j$ 为辛型.

辛型集合和 $\mathscr{R}(1,m)$ 在 $\mathscr{R}(2,m)$ 中的陪集集合之间存在一个一一映射. 因而 $\mathscr{R}(2,m)$ 恰好是 $\mathscr{R}(1,m)$ 的陪集并. 用符号 $\mathscr{B}$ 即表示一个辛型,也表示该辛型所对应的 $\mathscr{R}(1,m)$ 在 $\mathscr{R}(2,m)$ 中的陪集.

2. 重量分布

假设辛型 $\mathscr{B}$ 的秩是 $2h$,那么陪集 $\mathscr{B}$ 中码字的重量分布为 $A_{2^{m-1}-2^{m-h-1}}=2^{2h}$, $A_{2^{m-1}}=2^{m+1}-2^{2h+1}$, $A_{2^{m-1}+2^{m-h-1}}=2^{2h}$.

二阶 R-M 码 $\mathscr{R}(2,m)$ 的重量分布是: $A_0=A_{2^m}=1$, $A_i=0$, $i\neq 2^{m-1},2^{m-1}\pm 2^{m-1-h}$, $0\leqslant h\leqslant\lceil m/2\rceil$,

$$A_{2^{m-1}\pm 2^{m-1-h}}=2^{h(h+1)}\cdot\frac{(2^m-1)(2^{m-1}-1)\cdots(2^{m-2h+1}-1)}{(2^{2h}-1)(2^{2h-2}-1)\cdots(2^2-1)}.$$

3. 几类子码

假设 $\mathscr{S}$ 是包含具有这种性质的辛型的最大集合, $\mathscr{S}$ 中任意两个辛型之和的秩 $\geqslant 2d$, $1\leqslant d\leqslant\lceil m/2\rceil$. 记 $\mathscr{S}$ 对应的 $\mathscr{R}(2,m)$ 的子码是 S.

当 $m=2t+1$ 时,子码 S 是线性码,维数是 $m(t-d+2)+1$,最小距离是 $2^{m-1}-2^{m-d-1}$.

当 $m=2t+2\geqslant 4$ 时, S 是非线性码, 记为 $\mathscr{D}_{\mathscr{G}}(m,d)$, $2\leqslant d\leqslant t+1$. 它包含 $2^{(2t+1)(t-d+2)+2t+3}$ 个码字, 最小距离是 $2^{2t+1}-2^{2t+1-d}$. 特别地, 如果 $d=1$, 则 $\mathscr{D}_{\mathscr{G}}(m,1)=\mathscr{R}(2,m)$. 如果 $d=m/2=t=1$, 称 $\mathscr{D}_{\mathscr{G}}(m,d)$ 为 **Kerdock 码** $\mathscr{K}(m)$.

对于给定的 d, $\mathscr{R}(1,m)\subseteq\mathscr{K}(m)\subseteq\mathscr{D}_{\mathscr{G}}(m,d)\subseteq\mathscr{R}(2,m)$. 由于 $\mathscr{K}(m)$ 生成 $\mathscr{R}(2,m)$,所以 $\mathscr{D}_{\mathscr{G}}(m,d)$ 也生成 $\mathscr{R}(2,m)$.

非线性码 $\mathscr{K}(m)$ 的重量分布和距离分布是一样的, $A_0=A_{2^m}=1$, $A_{2^{m-1}}=$

$2^{m+1}-2$, $A_{2^{m-1}\pm 2^{(m-2)/2}} = 2^m(2^{m-1}-1)$.

Preparata码 $\mathscr{P}(m)$ 是一类非线性码，其重量分布(或距离分布)恰好是 $\mathscr{K}(m)$ 的重量分布的麦克威廉斯变换.因此,Preparata 码可以看作麦克威廉斯变换意义下 Kerdock 码的对偶.它的码字比具有相同参数线性码的码字多一倍.

9 二次剩余码

9.1 二次剩余码的定义

二次剩余码 $\mathscr{Q}$、$\overline{\mathscr{Q}}$、$\mathscr{N}$、$\overline{\mathscr{N}}$ 是有限素域 GF(l) 上特殊的循环码,其码长 p 为素数,素数 l 是模 p 的二次剩余.设模 p 的二次剩余的集合为 Q,非二次剩余的集合为 N,那么 Q 是模 p 的分圆陪集的并,多项式 $q(x)=\prod_{r\in Q}(x-\alpha^r)$ 和 $n(x)=\prod_{n\in N}(x-\alpha^n)$ 是多项式环 GF(l)[x] 中的元素,其中 α 是一个本原 p 次单位根,并且 $x^p-1=(x-1)q(x)n(x)$.二次剩余码 $\mathscr{Q}$、$\overline{\mathscr{Q}}$、$\mathscr{N}$、$\overline{\mathscr{N}}$ 的生成多项式分别是 $q(x)$、$(x-1)q(x)$、$n(x)$ 和 $(x-1)n(x)$.

显然,$\mathscr{Q}\supset\overline{\mathscr{Q}}$,$\mathscr{N}\supset\overline{\mathscr{N}}$.当 $l=2$ 时,$\overline{\mathscr{Q}}$ 是 $\mathscr{Q}$ 的偶重子码,$\overline{\mathscr{N}}$ 是 $\mathscr{N}$ 的偶重子码.

$\mathscr{Q}$ 和 $\mathscr{N}$ 是等价的,$\overline{\mathscr{Q}}$ 和 $\overline{\mathscr{N}}$ 是等价的,维数分别是 $(p+1)/2$ 和 $(p-1)/2$,最小距离大于等于 $p^{1/2}$.

记 $\hat{C}$ 是码 C 的扩展码,即添加一致奇偶校验得到的码.

当 $p=4k-1$ 时,$\mathscr{Q}$ 和 $\overline{\mathscr{Q}}$ 互为对偶,$\mathscr{N}$ 和 $\overline{\mathscr{N}}$ 互为对偶,$\hat{\mathscr{Q}}$ 和 $\hat{\mathscr{N}}$ 是自对偶码.

当 $p=4k+1$ 时,$\mathscr{Q}$ 和 $\overline{\mathscr{N}}$ 互为对偶,$\mathscr{N}$ 和 $\overline{\mathscr{Q}}$ 互为对偶,$\hat{\mathscr{Q}}$ 和 $\hat{\mathscr{N}}$ 互为对偶码.

9.2 二次剩余码的幂等元和生成矩阵

1. 二元二次剩余码

对于二元二次剩余码来说,码长 p 只能是 $8m\pm 1$ 的形式.

当 $p=8m-1$ 时,可以选择合适的本原 p 次单位根 α 使得 $\mathscr{Q}$、$\overline{\mathscr{Q}}$、$\mathscr{N}$、$\overline{\mathscr{N}}$ 的幂等元分别是

$$E_q(x)=\sum_{r\in Q}x^r,\ F_q(x)=1+\sum_{n\in N}x^n,\ E_n(x)=F_q(x)-1,\ F_n(x)=1+E_q(x).$$

当 $p=8m+1$ 时,可以选择合适的本原 p 次单位根 α 使得 $\mathscr{Q}$、$\overline{\mathscr{Q}}$、$\mathscr{N}$、$\overline{\mathscr{N}}$ 的幂等元分别是 $1+\sum_{r\in Q}x^r$, $\sum_{n\in N}x^n$, $1+\sum_{n\in N}x^n$, $\sum_{r\in Q}x^r$.

2. 非二元二次剩余码

设 $p = 4k \pm 1, \theta^2 = \begin{cases} p, & p = 4k+1, \\ -p, & p = 4k-1. \end{cases}$

那么 $\mathscr{Q}$、$\overline{\mathscr{Q}}$、$\mathscr{N}$、$\overline{\mathscr{N}}$ 的幂等元分别是

$$E_q(x) = \frac{1}{2}\left(1+\frac{1}{p}\right)+\frac{1}{2}\left(\frac{1}{p}-\frac{1}{\theta}\right)\sum_{r\in Q}x^r+\frac{1}{2}\left(\frac{1}{p}+\frac{1}{\theta}\right)\sum_{n\in N}x^n, \tag{9-1}$$

$$F_q(x) = \frac{1}{2}\left(1-\frac{1}{p}\right)-\frac{1}{2}\left(\frac{1}{p}+\frac{1}{\theta}\right)\sum_{r\in Q}x^r-\frac{1}{2}\left(\frac{1}{p}-\frac{1}{\theta}\right)\sum_{n\in N}x^n, \tag{9-2}$$

$$E_n(x) = \frac{1}{2}\left(1+\frac{1}{p}\right)+\frac{1}{2}\left(\frac{1}{p}+\frac{1}{\theta}\right)\sum_{r\in Q}x^r+\frac{1}{2}\left(\frac{1}{p}-\frac{1}{\theta}\right)\sum_{n\in N}x^n, \tag{9-3}$$

$$F_n(x) = \frac{1}{2}\left(1-\frac{1}{p}\right)-\frac{1}{2}\left(\frac{1}{p}-\frac{1}{\theta}\right)\sum_{r\in Q}x^r-\frac{1}{2}\left(\frac{1}{p}+\frac{1}{\theta}\right)\sum_{n\in N}x^n. \tag{9-4}$$

3. 生成矩阵

假设 $\overline{\mathscr{Q}}$ 的幂等元是 $F_q(x) = \sum_{i=0}^{p-1} f_i x^i$，则 $\overline{\mathscr{Q}}$ 的生成矩阵是一个秩为 $(p-1)/2$ 的轮换矩阵 $\overline{G} = (g_{ij}), 0 \leqslant i,j \leqslant p-1, g_{ij} = f_{j-i}$．$\mathscr{Q}$的生成矩阵是$\begin{bmatrix} & \overline{G} & \\ 1 & \cdots & 1 \end{bmatrix}$．同理，可以给出 $\mathscr{N}$和 $\overline{\mathscr{N}}$的生成矩阵．

9.3　二次剩余码的译码

本节叙述的两种译码算法不仅适用于二次剩余码，对其它一些线性码也是有效的．

假设二元(n,k)线性码 C 的最小距离是 d，生成矩阵 $G = [A \mid I]$，奇偶校验矩阵 $H = [I \mid A^{\mathrm{T}}]$．再设传送码字 $c = (c_0, c_1, \cdots, c_{n-1})$，差错向量 $e = (e_0, e_1, \cdots, e_{n-1})$ 的重量 $wt(e) = t \leqslant \lceil (d-1)/2 \rceil$，接收向量 $y = c + e = (y_0, y_1, \cdots, y_{n-1})$．那么 $c_0, c_1, \cdots, c_{r-1}$ 是 $r = n - k$ 个校验位，$c_r, \cdots, c_{n-1}$ 是 k 个信息位．

1. 置换译码法

对 C，首先构造置换的集合 $P = \{\pi_1 = 1, \pi_2, \cdots, \pi_s\}$ 使得 C 在每个 π_i 的作用下不变，并且对于每个重量不超过 t 的差错向量 e，存在 $\pi_j \in P$ 使得 e 在 π_j 的作用下所有的1都不在信息位．计算每个 πy 和伴随式 $S^{(i)} = H(\pi_i y)^{\mathrm{T}}$，直到找到一个置换 $\pi_i \in P$ 使得 $wt(S^{(i)}) \leqslant t$．那么差错发生在这个 $\pi_i y$ 的前 r 位，所以 $c = \pi_i^{-1}(\pi y + e_0, \cdots, e_{r-1}, 0, \cdots, 0)$．如果对所有的 $\pi_i \in P$ 都有 $wt(S^{(i)}) > t$，则有多于 t 个差错发生．

2. 覆盖多项式法

首先构造一个称为覆盖多项式的集合 $\{Q_0(x) = 0, Q_1(x), \cdots, Q_a(x)\}$ 使得对于任意一个重量不超过 t 的差错向量 e，存在一个 $Q_j(x)$ 满足在信息位上等于 e 的某个循环移位．令 Q_j 的伴随式是 $q_j = HQ_j^{\mathrm{T}}$．计算向量 y 的所有循环移位 $x^i y(x)$ 的

伴随式 $S^{(i)}$ 直到对某个 j 满足 $wt(S^{(i)}+q_j)\leqslant t-wt(Q_j(x))$. 那么 $c(x)=x^{n-i}(x^iy(x)+Q_j^{\mathrm{T}}+s^{\mathrm{T}}+q_j)$.

9.4 Golay 码

1. Golay 码的定义

不论从理论还是实际应用方面,Golay 码可能是最重要的码之一.

Golay 码指二元码 $\mathscr{G}_{23}$ 和 $\mathscr{G}_{24}$ 以及三元码 $\mathscr{G}_{11}$ 和 $\mathscr{G}_{12}$. $\mathscr{G}_{23}$ 是码长 $p=23$ 的二元二次剩余码,$\mathscr{G}_{24}$ 是 $\mathscr{G}_{23}$ 的扩展码;$\mathscr{G}_{11}$ 是码长 $p=11$ 的三元二次剩余码,$\mathscr{G}_{12}$ 是 $\mathscr{G}_{11}$ 的扩展码.

作为二次剩余码,$\mathscr{G}_{24}$ 和 $\mathscr{G}_{12}$ 都是自对偶码.

在同构意义下,Golay 码 $\mathscr{G}_{23}$、$\mathscr{G}_{24}$、$\mathscr{G}_{11}$ 和 $\mathscr{G}_{12}$ 都是唯一的.

$\mathscr{G}_{23}$ 和 $\mathscr{G}_{11}$ 是完全码.

用 A_i 表示码 C 中汉明重量是 i 的码字个数. 那么 $\mathscr{G}_{24}$ 的重量分布是

$$\begin{array}{rcccccc} i & : & 0 & 8 & 12 & 16 & 24 \\ A_i & : & 1 & 759 & 2576 & 759 & 1 \end{array} \tag{9-5}$$

$\mathscr{G}_{23}$ 的重量分布是

$$\begin{array}{rcccccccccc} i & : & 0 & 7 & 8 & 11 & 12 & 15 & 16 & 23 \\ A_i & : & 1 & 253 & 506 & 1288 & 1288 & 506 & 253 & 1 \end{array} \tag{9-6}$$

$\mathscr{G}_{12}$ 的重量分布是

$$\begin{array}{rcccccc} i & : & 0 & 6 & 9 & 12 \\ A_i & : & 1 & 264 & 440 & 24 \end{array} \tag{9-7}$$

2. Golay 码的自同构群

设 $\Omega=\{0,1,\cdots,22,\infty\}$, S、V、T、W 都是 Ω 上的置换. $S=(\infty)(0\ 1\ 2\cdots 22)$, $V=(\infty)(0)(1\ 2\ 4\ 8\ 16\ 9\ 18\ 13\ 3\ 6\ 12)(5\ 10\ 20\ 17\ 11\ 22\ 21\ 19\ 15\ 7\ 14)$, $T=(\infty\ 0)(1\ 22)(2\ 11)(3\ 15)(4\ 17)(5\ 9)(6\ 19)(7\ 13)(8\ 20)(10\ 16)(12\ 21)(14\ 18)$, $W=(\infty\ 0)(3\ 15)(1\ 17\ 6\ 14\ 2\ 22\ 4\ 19\ 18\ 11)(5\ 8\ 7\ 12\ 10\ 9\ 20\ 13\ 21\ 16)$,那么 Mathieu 群 M_{24} 定义为由置换 S、V、T、W 生成的群. Mathieu 群 M_{23} 定义为 M_{24} 中固定 Ω 中某个点的所有置换的集合,它是 M_{24} 的子群.

可以证明 Golay 码 $\mathscr{G}_{24}$ 和 $\mathscr{G}_{23}$ 的自同构群分别是 M_{24} 和 M_{23}.

设 $\Omega'=\{0,1,\cdots,9,10,\infty\}$, S'、V'、T'、$\overline{T'}$、Δ 都是 Ω' 上的置换, $S':i\to i+1$, $V':i\to 3i$, $\overline{T'}:i\to -1/i$, $\Delta=(\infty)(0)(1)(2\ 10)(3\ 4)(5\ 9)(6\ 7)(8)$, $T':i\to -\varepsilon/i$, 其中 $\varepsilon=\begin{cases}1, & i\in\{\infty,1,3,4,5,9\},\\ -1, & i\in\{0,2,6,7,8,10\}.\end{cases}$

Mathieu 群 M'_{12} 定义为由置换 S'、V'、T'、Δ 生成的群. Mathieu 群 M_{12} 定义为由置换 S'、V'、T'、Δ、生成的群. M'_{12} 和 M_{12} 是同构的,是 $\mathscr{G}_{12}$ 的自同构群.

Mathieu群定义为 M_{12} 中固定 Ω' 中某个点的置换的集合,它是 $\mathscr{G}_{11}$ 的自同构群.

10　代数几何码

代数几何码是近年来非常热门的课题.不论是代数几何学家还是编码专家都对代数几何码津津乐道.这不仅仅因为代数几何码是代数几何学与编码理论的完美结合,更重要的是可以构造出参数非常优良的代数几何码,达到甚至超过某些界.

设 X 是有限域 $GF(q)$ 上光滑绝对不可约的射影曲线,$K = GF(q)(X)$ 是 X 上的有理函数域,$\Omega(X)$ 是 X 上的有理微分 1- 形式空间,$X(GF(q))$ 是 X 的 $GF(q)$-点集,$\mathrm{Div}(X)$ 是 $GF(q)$- 除子群,$\mathrm{Div}^+(X)$ 是有效除子半群.$\mathrm{Pic}(X)$ 是除子类群,$J_X = \mathrm{Pic}^0(X)$ 是 X 的雅可比,即次数为零的除子类组成的代数群.

设 $D \in \mathrm{Div}(X)$,$\mathrm{Supp}(D)$ 表示 D 的支撑集,称向量空间 $L(D) = \{f \in K^* \mid (f) + D \geqslant 0\} \cup \{0\}$ 为 D 的相伴空间,$L(D)$ 是有限维的,其维数记为 $\mathscr{L}(D)$.定义 $\Omega(D) = \{\omega \in \Omega(X)^* \mid (\omega) + D \geqslant 0\} \cup \{0\}$.

10.1　代数几何码的构造

1. L 构造法

设曲线 X 的 $GF(q)$- 点集 $X(GF(q))$ 非空,令 $\mathscr{P} = \{P_1, P_2, \cdots, P_n\} \subseteq X(GF(q))$,$D \in \mathrm{Div}(X)$ 且 $\mathrm{Supp}(D) \cap \mathscr{P} = \varnothing$.考虑映射

$$\mathrm{Evp}: L(D) \to GF(q)^n,\quad Ev_{\mathscr{P}}(f) = (f(P_1), f(P_2), \cdots, f(P_n)),\qquad (10\text{-}1)$$

这样得到一个码 $\mathscr{C} = E_v\mathscr{P}(L(D))$,记为$(X, \mathscr{P}, D)_L$,称为**代数几何码**.

设 X 的亏格是 g,$0 \leqslant \deg D = a < n$,称 $k_{\mathscr{C}} = a - g + 1$,$d_{\mathscr{C}} = n - a$ 分别为代数几何码 C 的设计维数和设计距离.代数几何码$(X, \mathscr{P}, D)_L$ 的最小距离 $d \geqslant d_{\mathscr{C}}$,维数 $k = a - g + 1 + \mathscr{L}(K_X - D) - \mathscr{L}(D - P) \geqslant k_{\mathscr{C}}$,其中 $P = \sum_{P_i \in \mathscr{P}} P_i$,$K_X$ 是 X 的典范类.

2. Ω 构造法

我们知道,如果 $P \in X(GF(q))$,非零形式 ω 定义在 $GF(q)$ 上,则留数 $\mathrm{Res}_{\mathscr{P}}(\omega) \in GF(q)$.

令 $\mathscr{P} = \{P_1, P_2, \cdots, P_n\}$,$P = P_1 + \cdots + P_n \in \mathrm{Div}(X)$,考虑映射

$$\mathrm{Res}_{\mathscr{P}}: \Omega(P - D) \to GF(q)^n, \mathrm{Res}_{\mathscr{P}}(\omega) = (\mathrm{Res}\,P_1(\omega), \cdots, \mathrm{Res}P_n(\omega))\ (10\text{-}2)$$

这样构造了一个码 $\mathscr{C} = \mathrm{Res}_{\mathscr{P}}(\Omega(P - D))$,记为$(X, \mathscr{P}, D)_\Omega$.

假设 X 的亏格是 g,$\deg D = a$ 且 $2g - 2 < a$,称 $k_{\mathscr{C}} = n - a + g - 1$ 和 $d_{\mathscr{C}} = a - 2g + 2$ 分别是 $\mathscr{C}$的设计维数和设计距离.代数几何码$(X, \mathscr{P}, D)_\Omega$ 的最小距离 $d \geqslant d_{\mathscr{C}}$,维数 $k = n - a + g - 1 - \mathscr{L}(K_X - D) + \mathscr{L}(D - P) \geqslant k_{\mathscr{C}}$.

如果 $L(D)$ 的一组基是 $f_1, f_2, \cdots, f_m$,那么代数几何码$(X, \mathscr{P}, D)_\Omega$ 的奇偶校验矩阵是

$$\begin{bmatrix} f_1(P_1) & f_1(P_2) & \cdots & f_1(P_n) \\ f_2(P_1) & f_2(P_2) & \cdots & f_2(P_n) \\ \vdots & \vdots & & \vdots \\ f_m(P_1) & f_m(P_2) & \cdots & f_m(P_n) \end{bmatrix}.$$

对于 L 构造法和 Ω 构造法，都有 $d_{\mathscr{C}} + k_{\mathscr{C}} = n - g + 1$.

3. H 构造法

设 $\mathscr{L}$ 是 X 上的线丛，$H^0(\mathscr{L})$ 是其截口空间，$\overline{\mathscr{L}}_{P_i}$ 是 $\mathscr{L}$ 在 P_i 点的纤维，那么存在一个从 $H^0(\mathscr{L})$ 到 $\overline{\mathscr{L}}_{P_1} \oplus \cdots \oplus \overline{\mathscr{L}}_{P_n}$ 的自然映射 $\mathrm{Germ}_{\mathscr{P}}$. 又 $\overline{\mathscr{L}}_{P_1} \oplus \cdots \oplus \overline{\mathscr{L}}_{P_n}$ 和 $\mathrm{GF}(q)^n$ 同构，所以得到一个码 $\mathscr{C} = \mathrm{Germ}_{\mathscr{P}}(H^0(\mathscr{L}))$，记为 $(X, \mathscr{P}, \mathscr{L})_H$. 假设 $0 \leqslant \deg \mathscr{L} = a < n$，则 $(X, \mathscr{P}, \mathscr{L})_H$ 的维数 $k \geqslant a - g + 1$，最小距离 $d \geqslant n - a$.

4. 代数几何码的对偶

(1) $(X, \mathscr{P}, D)_L$ 和 $(X, \mathscr{P}, D)_\Omega$ 互为对偶码.

(2) 设 $P = \sum_{P_i \in \mathscr{P}} P_i$，$X$ 上线丛 $\tilde{\Omega}(P)$ 和 $\mathscr{L}$. 如果 $\mathscr{L}$ 和 $\tilde{\Omega}(P) \otimes \mathscr{L}^{-1}$ 有协合平凡化，则代数几何码 $(X, \mathscr{P}, \mathscr{L})_H$ 和 $(X, \mathscr{P}, \tilde{\Omega}(P) \otimes \mathscr{L}^{-1})_H$ 互为对偶码.

10.2　代数几何码的重量谱

设代数几何码 $\mathscr{C} = (X, \mathscr{P}, \mathscr{L})_H$ 的重量计数子 $W_{\mathscr{C}}(x : y) = x^n + \sum_{r=d}^{n} A_r x^{n-r} y^r$ 的非齐次化为 $W_{\mathscr{C}}(x)$. 则

$$W_{\mathscr{C}}(x) = x^n + \sum_{l=0}^{a} B_l (x-1)^l,$$

其中，当 $0 \leqslant l \leqslant a - 2g + 1$ 时

$$B_l = \binom{n}{l}(q^{a-l-g+1} - 1),$$

当 $a - 2g + 2 \leqslant l \leqslant a$ 时

$$\binom{n}{l}(q^{\lceil (a-l)/2 \rceil + 1} - 1) \geqslant B_l \geqslant \max\left\{0, \binom{n}{l}(q^{a-l-g+1} - 1)\right\}.$$

10.3　代数几何码的译码

设 $\mathscr{C} = (X, \mathscr{P}, D)_\Omega$，曲线 X 的亏格是 g，$2g - 2 < a = \deg D \leqslant n - g + 1$. 接收到向量 $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{e}$，$\boldsymbol{e}$ 是差错向量，$I = \{i \mid e_i \neq 0\}$ 是差错向量 $\boldsymbol{e}$ 的差错位的集合，$t = |I|$.

1. 基本算法

步 1　计算一个辅助除子 D'，使得 $\mathrm{Supp}(D) \cap \mathscr{P} = \varnothing$，$\deg D' = b$.

步 2　分别计算 $L(D)$, $L(D')$ 和 $L(D-D')$ 的基 $\{f_1,f_2,\cdots,f_m\}$, $\{g_1,g_2,\cdots,g_l\}$ 和 $\{h_1,h_2,\cdots,h_r\}$. 当然 $g_jh_k\in L(D)$.

步 3　对于接收的向量 $\boldsymbol{v}$,计算它的伴随式

$$S(\boldsymbol{v},f_i)=\underset{j\in I}{\mathrm{sum}}\, e_jf_i(P_j),\ S(\boldsymbol{v},g_jh_k)=s_{ij}=\sum_{i\in I}e_ig_jh_k(P_i).$$

步 4　计算线性方程组

$$\sum_{i=1}^{l}s_{ij}x_i=0,\quad j=1,2,\cdots,r. \tag{10-3}$$

的一个解 $\boldsymbol{y}=(y_1,\cdots,y_l)$.

步 5　定义 $g_y=\sum_{i=1}^{l}y_ig_i$. 找出所有使得 $g_y(P_i)=0$ 的 i 的集合 I_y,一定有

$$I_y\supseteq I.$$

步 6　线性方程组

$$\sum_{i\in I_y}f_j(P_i)z_i=S(\boldsymbol{v},f_j),\quad j=1,2,\cdots,m \tag{10-4}$$

的解就是差错向量 $\boldsymbol{e}$.

基本算法可以纠任意不超过 $t_0=\max\limits_{D'}\{\min\{L(D'),a-\deg D'-2g+2\}\}-1$ 个差错,这里 D' 是所有使得 $\mathrm{Supp}(D')\cap\mathscr{S}=\varnothing$ 成立的除子.

2. 修改的基本算法

这个修改的算法对于形如 $D=aP$ 的除子特别有效,P 是一个 Weierstrass 点.

设 $D=a_1H$, H 是一个次数为 h 的有效除子, $a=a_1h$. 定义

$$S(H)=\max_{i\in z}\left\{\left\lceil\frac{ih+h+1}{2}\right\rceil-\mathscr{L}(iH)\right\}.$$

步 1　取 $D'=H,2H,\cdots$,解方程组(10-3). 设 b_1 是使方程组(10-3)有解的最小整数,设解为 $\boldsymbol{y}$. 当 $t\leqslant\left\lceil\frac{d_{\mathscr{G}}-1}{2}\right\rceil-S(H)$ 时,线性方程组(10-4)有唯一解,这个解就是差错向量 $\boldsymbol{e}$.

后面的几步类似基本算法.

如果 $D=a_1H$, $H>0$,那么修改的基本算法可以纠任意不超过 $\left\lceil\frac{d_{\mathscr{G}}-1}{2}\right\rceil-S(H)$ 个差错. 当 X 是椭圆曲线或超椭圆曲线, $D=A=a_1H$ 是超平面截口,那么修改的基本算法可以纠任意不超过 $\left\lceil\frac{d_{\mathscr{G}}-1}{2}\right\rceil$ 个差错.

可以看出, $S(H)$ 越小,纠错能力就越强. 特别地,当 $S(H)=0$ 时,修改的基本算法可以纠 $\left\lceil\frac{d_{\mathscr{G}}-1}{2}\right\rceil$ 个差错. 然而 $S(H)=0$ 的情况很少出现,这是因为,如果 $S(H)=0$,那么或者亏格 $g\leqslant 1$,或者 H 是超椭圆除子, X 是超椭圆曲线.

一般地,

$$\lceil\frac{h-1}{2}\rceil \leqslant S(H) \leqslant \begin{cases} \lceil\frac{g+h}{2}\rceil - 1, & h \text{ 和} \lceil\frac{g-1}{h}\rceil \text{ 有一个是偶数}, \\ \lceil\frac{g+h-1}{2}\rceil, & h \text{ 和} \lceil\frac{g-1}{h}\rceil \text{ 都不是偶数}. \end{cases}$$

10.4 椭 圆 码

设 $\mathscr{C}$是一个(n,k)线性码,其对偶码为$(n-k,k)$线性码 $\mathscr{C}^{\perp}$,它们的最小距离分别是 d 和$d^{\perp}$,如果满足 $k+d \geqslant n+1-g$,$(n-k)+d^{\perp} \geqslant n+1-g$,则称$\mathscr{C}$是一个亏格最大为 g 的码.

亏格为 1 的曲线称为椭圆曲线,亏格最大为 1 的码称为**椭圆码**.

1. 椭圆码的重量谱

假设 X 是一椭圆曲线,$D = \sum a_i Q_i$ 是定义在 $\mathrm{GF}(q)$ 上的除子,那么存在一个点 $P_D \in X(\mathrm{GF}(q))$ 使得 $P_D = \sum a_i Q_i$,并且 P_D 仅仅依赖于除子D所在的除子类.因而,对于给定的线丛 $\mathscr{L}$,有一个点 $P_{\mathscr{L}}$与之对应.

符号 $M(X(\mathrm{GF}(q)),\mathscr{P},P_{\mathscr{L}},k)$ 表示集合$\{Q \subset \mathscr{P} \| Q| = k, P_Q = P_{\mathscr{L}}\}$ 中元素的个数.

椭圆码 $C = (X,\mathscr{P},\mathscr{L})_H$ 的计数子是

$$W_{\mathscr{C}}(x) = x^n + \sum_{i=0}^{k-1}\binom{n}{i}(q^{k-i}-1)(x-1)^i + B_k(x-1)^k, k+d \geqslant n,$$

其中 $$B_k = (q-1)M(X(\mathrm{GF}(q)),\mathscr{P},P_{\mathscr{L}},k).$$

特别地,当 $\mathscr{P} = X(\mathrm{GF}(q))$,$\gcd(k,n) = 1$ 时,$\mathscr{C}$的最小距离是 $n-k$,并且

$$B_k = \binom{n}{k}(q-1)/n.$$

当 $n = 2k$ 时,任意椭圆码都是形式上自对偶的.

2. 椭圆 MDS 码

满足 $k+d = n+1$ 的线性码称为 **MDS 码**,它是射影直线上码长为 $n \leqslant q+1$ 的代数几何码.

亏格最大为 1 的线性码是 MDS 的当且仅当 $B_k = 0$.

不存在码长超过 $q+1$ 的椭圆 MDS 码.

10.5 Hermitian 码

亏格为 g 的曲线X的 $\mathrm{GF}(q)$点数 $N_q(g) \leqslant q+1+\lceil 2q^{1/2}\rceil g$,使得等号成立的曲线为极大曲线.

由于代数几何码的设计参数 $k_{\mathscr{C}}+d_{\mathscr{C}} = n-g+1$,所以设计参数越好,曲线 X 的有理点就必须越多.

假设 $r = p^m$ 是素数幂,$q = r^2$,那么仿射方程为 $x^{r+1}+y^{r+1}+1 = 0$的 Hermitian

曲线 X_r 是极大光滑的平面曲线，亏格 $g = (r^2 - r)/2$，GF(q)-点数是 $r^3 + 1$. 作变量代换 $u = b/(y - bx)$，$v = xu - a$，$a, b \in \mathrm{GF}(q)$，$a^r + a = b^{r+1} = -1$，得到一个与 X_r 同构的曲线 $X'_r: v^r + v = u^{r+1}$.

令 $n = r^3$，m 是任意正整数，Q 是曲线 X'_r 的唯一无限极点，$\mathscr{P} = X'_r - Q$，q 元码 $\mathscr{C}_m$ 是代数几何码 $(X'_r, \mathscr{P}, m, Q)_L$ · $\overline{m} = tr + s \leqslant m, 0 \leqslant s \leqslant r - 1$，表示形如 $ir + j(r+1), i \geqslant 0, 0 \geqslant j \leqslant r - 1$ 的最大整数，记 $v(m) = 1 + t(t+1)/2 + \min(t, s)$. 则码 $\mathscr{C}_m$ 的最小距离 $d_m \geq r^3 - m$，维数 k_m 满足

$$k_m = \begin{cases} v(m), & m \leqslant r^2 - r - 2, \\ m + 1 - (r^2 - r)/2, & r^2 - r - 2 < m < r^3, \\ r^3 - v(r^3 + r^2 - r - 2 - m), & m \geqslant r^3. \end{cases}$$

$\mathscr{C}_m$ 和 $\mathscr{C}_{r^3 + r^2 - r - 2 - m}$ 互为对偶码.

可以证明 GF(2) 上的平面曲线 $E: y^2 + y = x^3 + x + 1$，GF(25) 上曲线 $E_1: y^2 + y = x^3$ 和 Klein 四次曲线 $X: x^3y + y^3z + z^3x = 0$ 都是极大的. 因而，可以类似构造 Hermitian 码的方法构造一些代数几何码.

10.6 模 码

1. 古典模曲线码

设整数环 $\mathbf{Z}$ 上的特殊线性群 $SL(2, \mathbf{Z}) = \left\{ \begin{pmatrix} a & c \\ b & d \end{pmatrix} \mid a, b, c, d \in \mathbf{Z}, ad - bc = 1 \right\}$ 的一个商群为 $\Gamma(1) = \mathrm{SL}(2, \mathbf{Z})/\{\pm 1\}$. 对于任意给定的正整数 N，定义 $\Gamma(1)$ 的两个同余子群 $\Gamma(N)$ 和 $\Gamma_0(N)$ 为

$$\Gamma(N) = \left\{ \begin{pmatrix} a & c \\ b & d \end{pmatrix} \middle| \begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} (\mathrm{mod} N) \right\},$$

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & c \\ b & d \end{pmatrix} \middle| \begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} (\mathrm{mod} N) \right\},$$

其中 * 表示 $\mathbf{Z}$ 中的任意元素.

对于 $\Gamma(N)$ 和 $\Gamma_0(N)$，分别存在唯一的复数域 $\mathbf{C}$ 上的光滑射影曲线 $X(N)$ 和 $X_0(N)$，使得把 $X(N)$ 和 $X_0(N)$ 看作黎曼(Riemann)曲面时分别同构于 $\Gamma(N) \setminus H^*$ 和 $\Gamma_0(N) \setminus H^*$，其中 $H = \{z \in \mathbf{C} \mid \mathrm{Im} z > 0\}$ 装备了通常的复拓扑，$\Gamma(N)$ 和 $\Gamma_0(N)$ 装备了离散拓扑，拓扑空间 $\Gamma(N) \setminus H$ 装备了黎曼曲面结构，$\Gamma(N) \setminus H^*$ 是 $\Gamma(N) \setminus H$ 的紧致化. 称 $X(N)$ 和 $X_0(N)$ 为**模曲线**.

设 ζ_N 是一个本原 N 单位根，$Q(\zeta_N)$ 是有理数域 $\mathbf{Q}$ 添加 ζ_N 得到的扩域，K_N 是其唯一的二次子域. 根据椭圆曲线的模模型理论，对于给定的素数 p，$\gcd(p, N) = 1$，存在一个有限域 GF(p) 上光滑绝对不可约的射影曲线 $X_N = X_0(N)/p$ 是 $X(N)$ 的优约化；而且，对于域 K_N 中一个整除 p 的素理想 $\mathscr{C}$，存在一个域 $k(\mathscr{C}) \subseteq \mathrm{GF}(p^2)$ 上的光滑绝对不可约的射影曲线 $X'_N = X_p(N)/\mathscr{C}$ 是 $X_p(N)$ 的优约化. 实际上，当 $\mathscr{C} = (p)$ 时，$k(\mathscr{C})$ 同构于 GF(p^2)，当 $(p) = \mathscr{C}\mathscr{C}'$ 时，$\mathscr{C}'$ 是 $\mathscr{C}$ 的复共轭理想，

$k(\mathscr{C})$ 同构于 $GF(p)$.

令曲线 X_N 和 X'_N 的亏格分别是 $g_N = g(X_N) = g_0(N)$ 和 $g'_N = g(X'_N) = g(N)$,则

$$g(N) = 1 + N^2(N-6)/24\prod_l(1-l^{-2}),$$

l 是 N 的所有素因子,

$$g_0(N) = 1 + \mu/12 - v_2/4 - v_3/3 - v_\infty/2,$$

其中 $\mu = N\prod_l(1+l^{-1})$,

$$v_2 = \begin{cases} 0, & 4 \mid N, \\ \prod_l\left(1+\left(-\frac{1}{l}\right)\right), & 否则, \end{cases}$$

$$v_3 = \begin{cases} 0, & 9 \mid N \\ \prod_l\left(1+\left(-\frac{3}{l}\right)\right), & 否则, \end{cases}$$

$$v_\infty = \sum_{d \mid N, d>0}\varphi(d, N/d),$$

$$\left(-\frac{1}{l}\right) = \begin{cases} 0, & l = 2, \\ 1, & l \equiv 1 \pmod 4, \\ -1, & l \equiv 3 \pmod 4, \end{cases}$$

$$\left(-\frac{3}{l}\right) = \begin{cases} 0, & l = 3, \\ 1, & l \equiv 1 \pmod 3, \\ -1, & l \equiv 2 \pmod 3, \end{cases}$$

φ 是欧拉函数.

设 $\mathscr{P}$ 表示 X_N 的超奇异点集合,$\mathscr{P}'$ 表示 X'_N 的超奇异点集合,则 $\mathscr{P} \subseteq X_N(GF(p^2))$,$\mathscr{P}' \subseteq X'_N(GF(p^2))$.再设 D 是定义在 $GF(p^2)$ 上 X_N 的一个次数是 a 的除子,并且 $\mathrm{Supp}(D) \cap \mathscr{P} = \varnothing$.那么 $GF(p^2)$ 上的代数几何码 $(X_N, \mathscr{P}, D)_L$ 的码长 $n \leqslant n_0$,维数 $k \geqslant a - g_0(N) + 1$,最小距离 $d \geqslant n - a$,其中 $n_0 \geqslant \frac{N(p-1)}{12}\prod_l(1+l^{-1})$,$l$ 是 N 的所有素因子.

特别地,当 $N = 2^m$,$D = a\cdot\infty$,点 ∞ 在 $X_0(N)$ 上,$\deg D = a$ 时,$n_0 \geqslant 2^{m-3}(p-1)$,$k \geqslant k(m,a)$,$d \geqslant n - a$,其中 $k(m,a) = l$,当且仅当 $2^{m-3} - 2^{m-2-l} \leqslant a \leqslant 2^{m-3} - 2^{m-3-l}$,$l = 1,2,\cdots,\lceil m/2\rceil - 1$.

再设 D' 是定义在 $GF(p^2)$ 上 X'_N 的一个次数是 a' 的除子,并且 $\mathrm{Supp}(D') \cap \mathscr{P}' = \varnothing$.那么 $GF(p^2)$ 上的代数几何码 $(X'_N, \mathscr{P}', D')_L$ 的码长 $n' \leqslant n'_0$,维数 $k \geqslant a' - g(N) + 1$,最小距离 $d' \geqslant n' - a'$,其中 $n'_0 \geqslant \frac{N^3(p-1)}{24}\prod_l(1-l^{-2})$,$l$ 是 N 的所有素因子.

2. Drinfeld 曲线码

设 C 是一维射影空间.$A = GF(q)[T]$,$k = GF(q)(C)$ 表示 A 的分式域,k_∞ 表

示 k 的 $|\cdot|_\infty$ 完备化.$(0) \neq I$ 是 A 的真理想,P_I 是生成 I 的首一多项式,$G = (L,\varphi)$ 是 A 概型 S 上的一个椭圆模,λ 是一个 I 水平的结构.定义 $F(I)(S) = \{(G,\lambda)$ 的同构对集合$\}$.那么由表示理论可知,函子 $F(I)$ 可以由 A 上有限型的仿射概型 $M(I)$ 表示,并且存在唯一的概型 $\overline{M(I)}$ 使得 $M(I)$ 在 $\overline{M(I)}$ 中是开的稠密子概型,$\overline{M(I)}$ 在 $\mathrm{Spec}A[I^{-1}]$ 上的纤维是光滑完备曲线.$\overline{M(I)} \otimes_A k_\infty$ 称为 Drinfeld 模曲线.

再假设理想 I 是素的且满足 P_I 是不可约,$\deg P_I = m$ 是奇数,$\gcd(m, q-1) = 1$.人们最感兴趣的是光滑完备曲线 $X(I) = \overline{M(I)} \otimes_A \mathrm{GF}(q)$,它是可约的.

设一般线性群 $\mathrm{GL}(2,A)$ 的子群

$$\Gamma_0(I) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{GL}(2,A) \mid c \in I \right\},$$

$$\Gamma_1(I) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{GL}(2,A) \mid c \in I, a - 1 \in I \right\},$$

$$H(I) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{GL}(2,A) \mid b \in I, c \in I, a - d \in I \right\}$$

在 $\mathrm{GF}(2, A/I)$ 中的像分别是 $\overline{\Gamma_0(I)}$, $\overline{\Gamma_1(I)}$ 和 $\overline{H(I)}$,定义

$$X_0(I) = X(I)/\overline{\Gamma_0(I)},\quad X_1(I) = X(I)/\overline{\Gamma_1(I)},\quad C(I) = X(I)/\overline{H(I)},$$

那么 $X_0(I)$、$X_1(I)$ 和 $C(I)$ 都是光滑绝对不可约的完备曲线.

设 $\mathscr{P}$ 是曲线 $C(I)$ 的超奇异 $\mathrm{GF}(q^2)$ 有理点的集合,$\mathscr{G}$ 是一个次数为 a 的线丛,那么 $\mathrm{GF}(q)$ 上代数几何码$(C(I), \mathscr{P}, \mathscr{G})_H$ 的码长 $n = q^m(q^{2m}-1)/(q+1)$,维数 $k \geq a - (q^{2m}-1)(q^m-1)(q^m-q-1)/(q^2-1)$,最小距离 $d \geq q^m(q^{2m}-1)/(q+1) - a$.这里 $m = \deg P_I$.

设 $\mathscr{P}'$ 是曲线 $X_0(I)$ 的超奇异点集合,那么码$(X_0(I), \mathscr{P}', \mathscr{G})_H$ 的码长 $n = (q^m+1)/(q+1)$,维数 $k \geq a - q(q^{2l}-1)/(q^2-1)+1$,最小距离 $d \geq (q^m+1)/(q+1) - a$,这里 $l = \lceil m/2 \rceil$.

最后指出,不论是古典模曲线码,还是 Drinfeld 曲线码,都存在多项式算法构造它们的生成矩阵.

参 考 文 献

1 杨义先等.编码密码学.北京:人民邮电出版社,1992.

2 Macwilliams F J, Sloane N J. The theory of error-correcting codes, New York: North-Holland Pub, 1977.

3 Tsfasman M A, Vladut S G. Algebraic-geometric codes, Boston: Kluwer Academic Publishers, 1991.

4 林舒,科斯特洛著.王育民,王新梅译.差错控制编码.北京:人民邮电出版社,1986.

5 王新梅编著.纠错码与差错控制.北京:人民邮电出版社,1989.

·计算机数学卷·

第23篇

近代密码学

编　者　郭宝安
审校者　戴一奇

目　录

引　　言

当今人类已经进入信息社会,如何对信息资源进行安全的保护和管理,是我们所面临的重要课题,采用密码技术是保护信息安全的主要手段之一.密码技术自古有之,到目前为止,已经从外交和军事领域走向公开,它并且是结合数学、计算机科学、电子与通信等诸多学科于一身的交叉学科.它不仅具有保证信息机密性的信息加密功能,而且具有数字签名、身份验证、秘密分存、系统安全等功能,所以,使用密码技术不仅可以保证信息的机密性,而且可以保证信息的完整性和确证性,防止信息被篡改、伪造和假冒.密码技术除了在军事、外交等领域发挥作用以外,还在政府部门、银行、财政、税务、海关等电子化系统中发挥独特的作用.

1　密码学术语与概念

1.1　基本概念

密码学包括密码编码学和密码分析学.密码体制的设计是密码编码学的主要内容,密码体制的破译是密码分析学的主要内容.密码编码技术和密码分析技术是相互依存、互相支持、密不可分的两个方面.对于消息 m,一般的加密和解密过程如图 1-1 所示:

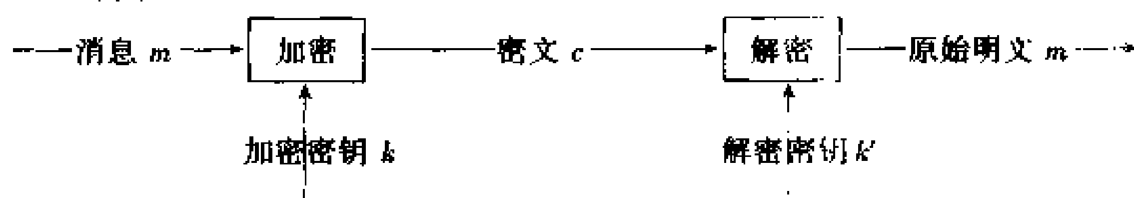


图 1-1

加密过程为 $c=E(m,k)$,其中 m 为明文,k 为加密密钥,E 为加密算法,c 为密文;解密过程为 $m=D(c,k')$,其中 D 为解密算法,k' 为解密密钥,E 和 D 可以相同,也可以不同.

从密码体制方面而言,密码体制有**对称密钥密码体制**(传统体制)和**非对称密钥密码体制**(公开密钥体制).对称密钥密码体制要求加密解密双方拥有相同的密钥($k=k'$)或 k 与 k' 之间通过计算容易相互得到.而非对称密钥密码体制要求加密解密双方拥有不相同的密钥($k\neq k'$),在不知道陷门信息的情况下,加密密钥和解密密钥在计算上是不能相互算出的.

密码分析的基本原则是假定破译者知道密码算法，但不知加密用的密钥.

(1) **唯密文攻击**　破译者仅知道 $c_1 = E(m_1,k), c_2 = E(m_2,k), \cdots, c_n = E(m_n,k)$，破译者推导出 $m_1, m_2, \cdots, m_n$ 或 k 或找出一个算法从 $c_{n+1} = E(m_{n+1},k)$ 中推导出 m_{n+1}.

(2) **已知明文攻击**　破译者知道有限多的明密文对：$(m_1, c_1 = E(m_1,k))$，$(m_2, c_2 = E(m_2,k)), \cdots, (m_n, c_n = E(m_n,k))$，破译者推导出 k 或找出一个算法从 $c_{n+1} = E(m_{n+1},k)$ 中推导出 m_{n+1}.

(3) **选择明文攻击**　破译者知道 $c_1 = E(m_1,k), c_2 = E(m_2,k), \cdots, c_n = E(m_n,k)$，其中 $m_1, m_2, \cdots, m_n$ 是破译者自己选择的，破译者推导出 k 或找出一个算法从 $c_{n+1} = E(m_{n+1},k)$ 中推导出 m_{n+1}.

(4) **自适应选择明文攻击**　破译者不仅能够选择被加密的明文，而且可以根据以前的加密结果修正以后的明文选择. 它是选择明文攻击的特殊情况.

(5) **选择密文攻击**　破译者能选择不同的密文，得到相应的明文，要求破译者推导出密钥.

(6) **选择密钥攻击**　破译者选择不同的密钥，通过密钥的相关性，根据选择明文或已知明文来求出密钥.

1.2　信息论基础

仙农(Shannon)在其关于密码学的理论中引入的以下原则和概念，直到今天对密码理论仍具有重要意义.

混乱(confusion)：用于掩盖明文与密文之间的关系，使得明文与密文之间的关系复杂化，目的是挫败通过研究密文的多余度和统计攻击. 代替是其主要手段.

散布(diffusion)：将明文的多余度充分分散到密文中. 主要方法是通过换位.

密码体制的**熵** $H(K)$ 定义为 $H(K) = \mathrm{lb}N(K)$，$N(K)$ 是密钥数目.

在密码系统的设计中要充分利用混乱和散布方法.

定理 1　对一个长度为 n 的消息，将一个密文解密为同一语言中的原始明文，采用不同密钥的数目为 $2^{H(K)-nD} - 1$，其中 D 为语言的多余度.

对于英语，$D = 3.5$ 比特 / 字母.

唯一解距离被定义为 $U = H(K)/D$. 一个密码体制的唯一解距离越大，其强度就越高. 唯一解距离的意义在于当获得的密文数目大于唯一解距离时，从理论上就能唯一地确定明文. 但是对许多实际的密码系统，在计算上还没有相应的算法来破译它们.

加密解密双方每加密一个消息，仅使用一个密钥，密钥每次更换，永不重复，密钥空间无限大的密码体制被称为**一次一密密码体制**. 它是唯一一种完全理想的保密体制，在理论上是不可破的，其余的所有体制在理论上都是可破的. 但理论上不可破的密码不是实用的.

密码学不仅仅是编码与破译的学问，而且包括安全协议设计、秘密分存、散列函数等内容. 总而言之，近代密码是一支引人入胜的领域，它涉及到数学的许多分

支,如数论、复杂性理论、信息论、概率统计、代数理论,甚至于代数几何等都在密码学的发展中留下各自的贡献.

2 对称密钥密码技术

2.1 序列密码

序列密码一直是作为军事和外交场合使用的主要密码技术之一,它的主要原理是,通过有限状态机产生性能优良的伪随机序列,使用该序列加密信息流(逐比特加密),得到密文序列,如图 2-1 所示.所以,序列密码算法的安全强度完全取决于它所产生的伪随机序列的好坏.

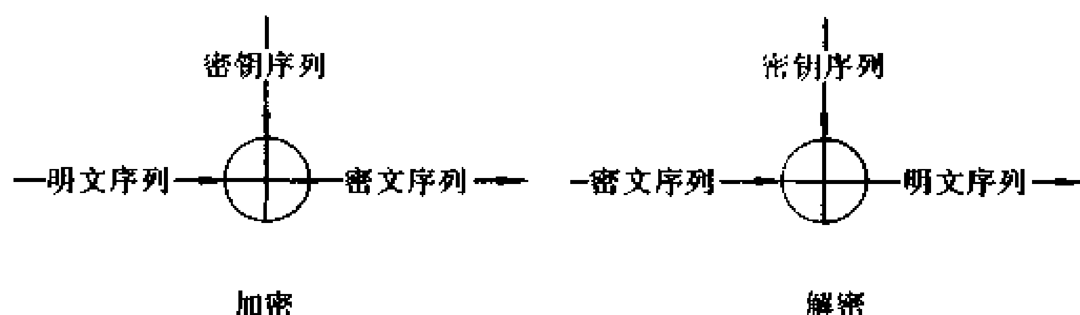


图 2-1

设明文序列为 $m_1, m_2, \cdots, m_n$,密钥序列为 $k_1, k_2, \cdots, k_n$,则密文 $c_i = m_i \oplus k_i$.其中 $\oplus$ 为 mod 2 加.

关于 0-1 序列的随机性公设如下:

1° 若序列周期 r 是奇数,则一个周期内 0 的个数比 1 的个数多一个或少一个;若 r 是偶数,则 0 和 1 的个数相等.

2° 序列中连续由 0 或 1 组成最长的子串,称为 **0 游程**或 **1 游程**.1 游程的数目为游程总数的 1/2,2 游程的个数占游程总数的 $1/2^2$,…c 游程的数目为总数的 $1/2^c$.而且对任意长度的 0 游程和 1 游程数目相同.

3° 假定序列为 $s_1, s_2, \cdots, s_r, \cdots$.对于下列两个子序列

$$s_1, s_2, \cdots, s_r; \qquad s_{1+\tau}, s_{2+\tau}, \cdots, s_{r+\tau},$$

$s_i = s_{i+\tau}$ 的数目用 n_τ 表示,不同的数目为 $d_\tau = r - n_\tau$,

$$R(\tau) = \frac{(n_\tau - d_\tau)}{r}$$

是一常数.

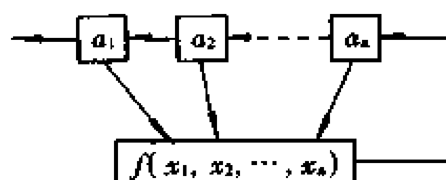


图 2-2

一般采用移位寄存器来产生伪

随机序列,如图 2-2 所示,即

$$a_n(t+1)=f(x_1(t),\ x_2(t),\cdots,x_n(t)),$$
$$a_{k-1}(t+1)=a_k(t),\ k=2,3,\cdots,n$$

已知 $a_1(0),a_2(0),\cdots,a_n(0)$.

当 $f(x_1,x_2,\cdots,x_n)$ 为线性(非线性)函数时,称为 n 级线性(非线性)反馈移位寄存器,即当 $a_n(t+1)=c_na_1(t)\oplus c_{n-1}a_2(t)\oplus\cdots\oplus c_1a_n(t)$ 时,为线性反馈移位寄存器,其中 $c_i=0$ 或 $1,i=1,2,\cdots,n$, $\oplus$ 为并或运算,即

$$0\oplus 0=0,\ 0\oplus 1=1\oplus 0=1,\ 1\oplus 1=0.$$

当 n 级反馈移位寄存器的反馈函数为线性函数,即

$$f(x_1,x_2,\cdots,x_n)=c_nx_1+c_{n-1}x_2+\cdots+c_1x_n$$

时,可将其用多项式表示:

$$f(x)=1+c_1x+c_2x^2+\cdots+c_nx^n.$$

称 $f(x)$ 为线性移位寄存器的联接多项式.

定义 1 二元域 F_2 上 n 级线性反馈移位寄存器产生的周期为 2^n-1 的非0序列,称为 n 级最大长线性反馈移位寄存器序列,简称 **m 序列**;以非线性反馈函数产生的周期为 2^n 的非0序列,称为**全长序列或 M 序列**.

关于线性反馈移位寄存器产生的 m 序列有以下性质(证明从略):

(1) 线性反馈移位寄存器产生 m 序列的充分必要条件是其反馈函数为 F_2 上的本原多项式.

(2) n 级互不平移等价的 m 序列的个数为 $\phi(2^n-1)/n$.

(3) m 序列满足以上 0-1 序列随机性公设的三个条件.

(4) n 级 M 序列的总数为 $2^k,k=2^{n-1}-n$.

定义 2 给定序列 a,能够产生 a 的最短的线性反馈移位寄存器的级数,称为该序列的**线性复杂度**.用 $L(a)$ 表示.

给定长度为 N 的序列 $a=(a_0,a_1,\cdots,a_{N-1})$,计算产生 a 的最短的线性反馈移位寄存器的级数 l_N 和反馈多项式 $f_N(x)$,有梅西算法如下:

对 $n=1,2,\cdots,N$,记 $(f_n(x),l_n)$ 为产生 $a_0,a_1,\cdots,a_{n-1}$ 的反馈多项式和级数.

(1) 取初值 $f_0(x)=1,l_0=0$.

(2) 设 $(f_n(x),l_n)$ 已求得,$N>n\geqslant 0$,而 $l_0\leqslant l_1\leqslant\cdots\leqslant l_n$,记

$$f_n(x)=c_0+c_1x+c_2x^2+\cdots+c_{l_n}x^{l_n},\quad c_0=1,$$

计算 $d_n=c_0a_n+c_1a_{n-1}+\cdots+c_{l_n}a_{n-l_n}$.称 d_n 为第 n 步差值.

如果 d_n 为 0,则 $f_{n+1}(x)=f_n(x),l_{n+1}=l_n$;

如果 d_n 为 1,则

1) 当 $l_0=l_1=\cdots=l_n=0$ 时,$f_{n+1}(x)=1+x^{n+1},l_{n+1}=n+1$;

2) 当有 $m(0\leq m<n)$ 使 $l_m<l_{m+1}=\cdots=l_n$ 时,则

$$f_{n+1}(x)=f_n(x)+x^{n-m}f_m(x),$$
$$l_{n+1}=\max(l_n,n+1-l_n).$$

最后得到的$(f_N(x), l_N)$便是产生序列 a 的最短线性反馈移位寄存器.

一个在密码系统中可用的序列,除了满足前面提到的三个条件以外,还要满足线性复杂度随机走动条件,它要求序列的线性复杂度是其长度的一半左右,详细可见有关资料的随机序列线性复杂度均值与方差部分.

设计用于密码系统的伪随机序列,还必须详细考察所用的布尔函数的密码特性.布尔函数 $f(x_1, x_2, \cdots, x_n)$ 是平衡的,是指其真值表中的0、1个数相等.此外,对布尔函数还定义了其它较复杂的概念,主要有:k 阶雪崩函数、m 阶相关免疫函数、Bent 函数等,这些函数不仅对设计序列密码有用,而且在分组密码设计中也十分有用.详细参见有关文献.

伪随机序列主要利用移位寄存器来产生,典型方法有:

(1) **反馈移位寄存器序列** 采用 n 阶非线性反馈函数产生大周期的非线性序列,例如 M 序列.它具有较好的密码学性质,只是反馈函数的选择有难度,产生全部的 M 序列至今仍是很复杂.

(2) **前馈序列** 利用线性移位寄存器序列加非线性前馈函数,产生前馈序列.如何控制序列相位及非线性前馈函数也是相当复杂的问题,Bent 序列就是其中一类好的序列.一般要求产生前馈序列的前馈布尔函数满足相关免疫条件等.

(3) **钟控序列** 利用一个寄存器序列作为时钟控制另一寄存器序列(或自己控制自己)来产生钟控序列.这种序列具有大的线性复杂度,但是相关性能较差.

(4) **组合网络及其它序列** 通过组合运用以上方法,产生更复杂的网络,来实现复杂的序列.这种序列的密码性质理论上比较难控制.

(5) 利用混沌理论、细胞自动机等方法产生的伪随机序列.

序列密码的优点是错误扩展小,速度快,利于同步,安全程度高.破译序列密码的方法主要有代数方法和统计方法,以及线性逼近和相关攻击等.

2.2 分组密码

分组密码的工作方式是将明文分成固定长度的组(块),如 64 比特一组,用同一密钥和算法对每一块加密,输出也是固定长度的密文.设计分组密码算法的核心技术是:在相信复杂函数可以通过简单函数迭代若干圈得到的原则下,利用简单圈函数及对合运算,充分利用非线性运算,通过多次迭代充分实现混乱与散布.

2.2.1 DES 算法(美国数据加密标准,1977 年颁布)

DES密码算法的输入为64比特明文,密钥长度64比特(含8比特奇偶校验位),密文长度 64 比特.DES 算法的主要步骤为:

(1) 将明文 m 经过初始置换 IP 的结果的前 32 位计为 L_0,后 32 位计为 R_0.

(2) 根据以下规则迭代16次:计算 $L_i = R_{i-1}, R_i = l_{i-1} \oplus f(R_{i-1}, k_i), i = 1,2, \cdots, 16$,"$\oplus$"为两个 32 位串的逐比特异或,$f$ 为一非线性函数,k_i 为 48 比特的子密钥,共 16 个.

(3) 对比特串 $R_{16}L_{16}$ 使用逆置换 IP^{-1},得到密文 y.

DES 算法总体流图如图 2-3.

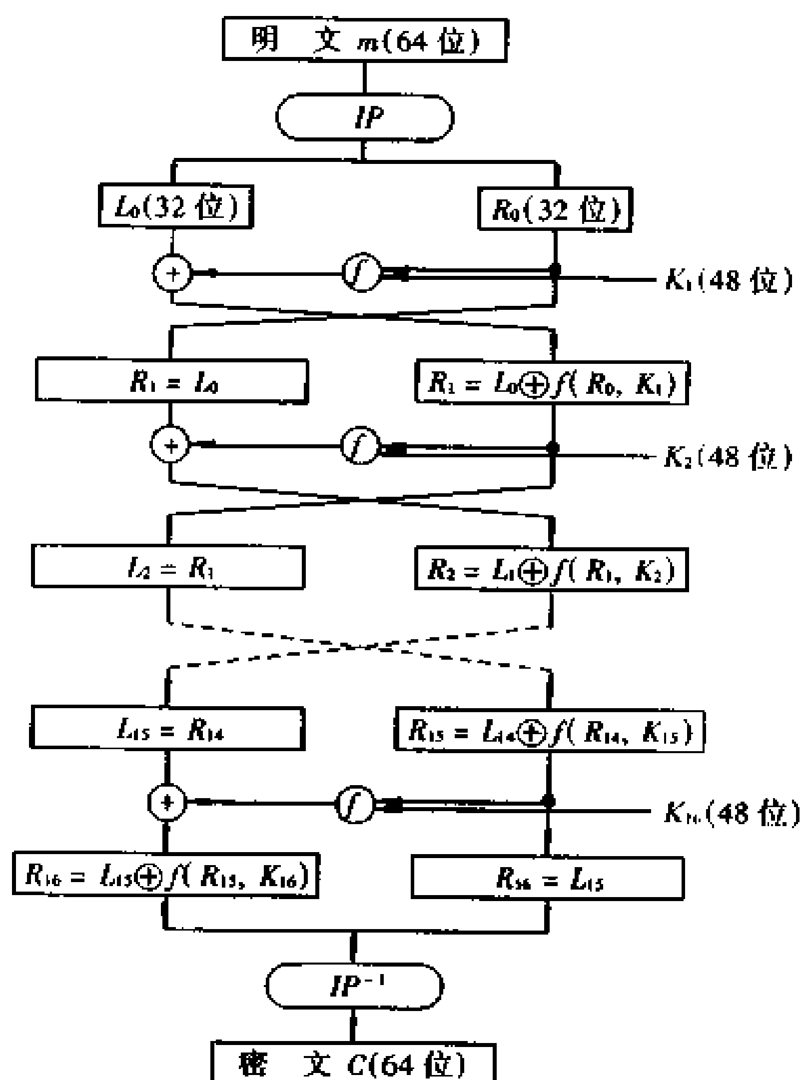


图 2-3

IP 为

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1} 为

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

假定 $m = m_1 m_2 \cdots m_{64}$, $IP(m) = a_{58} a_{50} a_{42} \cdots a_{23} a_{15} a_7$. 以后同此不多叙述.

子密钥 k_i 的产生过程:

设 $k = k_1 k_2 \cdots k_{64}$,

(1) 对64比特的密钥 k,去掉其8比特的奇偶校验位(即 $a_8, a_{16}, a_{24}, \cdots, a_{64}$ 为奇偶校验位),再使用PC-1置换得到56比特串,即 $PC\text{-}1(k) = C_0 D_0$, C_0 为前28位, D_0 为后28位. 其中 PC-1 为

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(2) 对 $i = 1, 2, \cdots, 16$, $C_i = LS_i(C_{i-1})$, $D_i = LS_i(D_{i-1})$, LS表示循环左移,当 $i = 1, 2, 9, 16$ 时,移位量为1,其它情况为2.

(3) $k_i = PC\text{-}2(C_i D_i)$,其中PC-2为

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

k_i 为48比特的子密钥.

f 函数:

函数 f 的输入为一个32比特串 A 和一个48比特的子密钥串 J,输出为一个32比特串. 其详细过程如下:

(1) 将 A 按照扩展变换 E 变为长为48比特的串 $E(A)$、E 为

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(2) 计算 $E(A) \oplus J$,并将结果写成连续的 8 个 6 比特串,记为

$$B = B_1 B_2 \cdots B_8$$

(3) 对每个 $B_j, j = 1,2,\cdots,8$,用以下 8 个 S 盒中相应的 $S_j (j = 1,2,\cdots,8)$ 做置换,将 6 比特信息变为 4 比特,即 $C_j = S_j(B_j)$,具体置换步骤为:

如 $B_j = b_1 b_2 \cdots b_6, r = b_1 b_6, c = b_2 b_3 b_4 b_5$,对 S_j 查表,其第 r 行、第 c 列就是置换结果.总共 32 比特,记为 C.

8 个 S 盒置换为

S 盒 1:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S 盒 2:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S 盒 3:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S 盒 4:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S 盒 5:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S 盒 6:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S 盒 7:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S 盒 8:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

(4) 将 C 进行 P 置换,得到最后的 32 比特即为 $f(A,J)$、P 置换为

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

例1 若明文为0123456789ABCDEF(16进制),密钥为133457799BBCDFF1,则密文为85E813540F0AB405.

DES的密钥有弱密钥、半弱密钥和互补密钥,选择密钥时要注意这些问题,避免使用这些密钥.DES受到的最大攻击是它的密钥长度仅有56比特,强力攻击的代价随着电子技术的突飞猛进越来越低,1990年比汉(S.Biham)和沙米尔(A.Shamir)提出了差分攻击的方法,采用选择明文2^{47}攻击,最终找到可能的密钥.麦特苏(M.Matsui)提出的线性分析方法,利用2^{43}个已知明文,成功地破译了16圈DES算法.这些都是密码分析上的重要成果.虽然只有理论的价值.在Internet上采用穷举搜索方法已成功地破译了56比特密钥的DES算法.

基于以上弱点,人们将DES算法作了多种变形,使用三重DES方式.独立子密钥方法、其理论依据是DES对56比特的加密运算不构成群.也有使用可变的S盒及其使用次序以及广义的GDES等技术的.这些改变有些是增强了密码算法的安全性,有些作用不大,有些还削弱了DES的安全性,详细可见参考文献.

鉴于DES的脆弱性,美国政府颁布了新的数据加密系统EES,该系统采用的密码算法为Skipjack算法,不对外公开,密钥长度80比特,封存在Clipper芯片中.为了便于政府监管,还设计了具有第三方托管的安全协议,在法律许可的条件下,政府可以对进行保密通信的密码进行解密监听.

2.2.2 IDEA(国际数据加密算法)

IDEA算法总体流图如图2-4.

其中$\boxplus$为模2^{16}加法运算,$\odot$为模$2^{16}+1$的乘法运算,$\oplus$为逐比特异或运算.算法的执行过程为:

将64比特的明文分成4个16比特的子块:x_1,x_2,x_3,x_4,它们作为第一轮的输入,与6个16比特的子密钥进行运算.运算顺序为:

(1) 将x_1与第一轮的第一个子密钥进行$\odot$运算.

(2) 将x_2与第一轮的第二个子密钥进行$\boxplus$运算.

(3) 将x_3与第一轮的第三个子密钥进行$\boxplus$运算.

(4) 将x_4与第一轮的第四个子密钥进行$\odot$运算.

(5) 将(1)与(3)的结果相异或,进行$\oplus$运算.

(6) 将(2)与(4)的结果相异或,进行$\oplus$运算.

(7) 将第(5)步的结果与第一轮的第五个子密钥进行$\odot$运算.

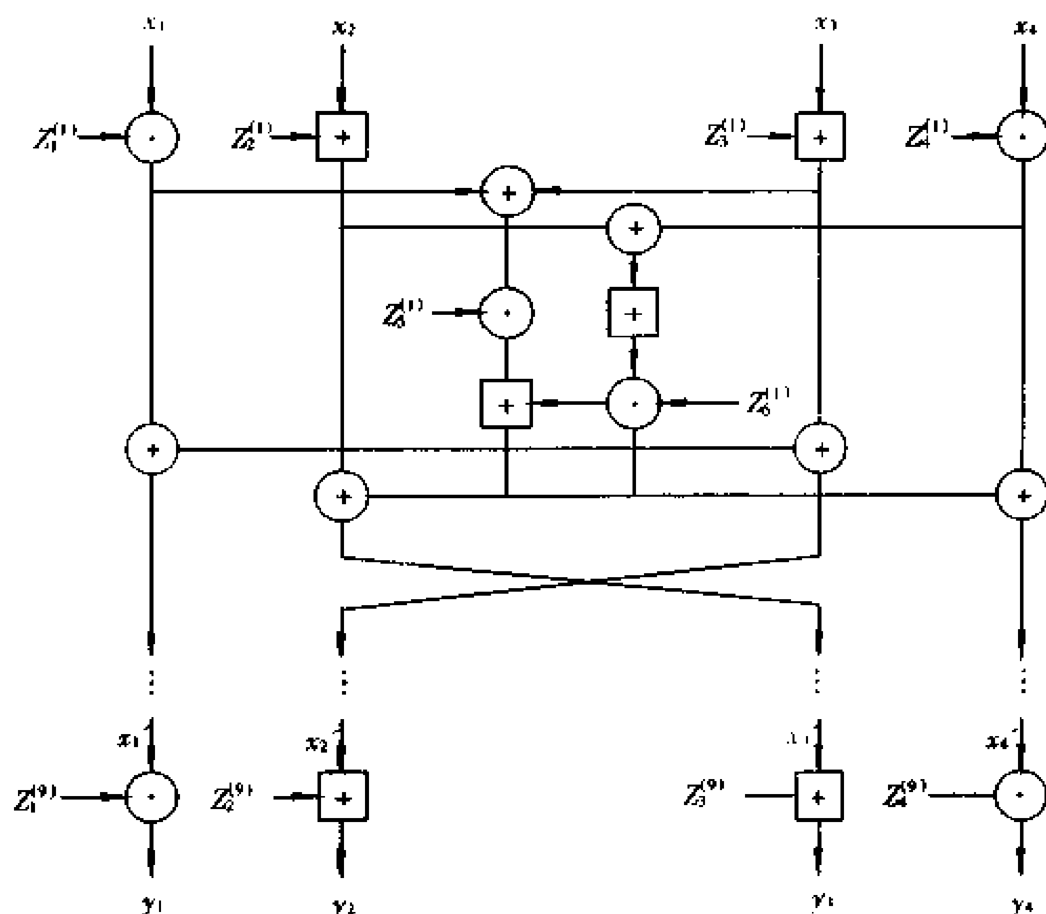


图 2-4

(8) 将第(6) 步和第(7) 步的结果进行 ⊞ 运算.

(9) 将第(8) 步的结果和第一轮的第六个子密钥进行 ⊙ 运算.

(10) 将第(7) 步和第(9) 步的结果进行 ⊞ 运算.

(11) 将第(1) 步和第(9) 步的结果相异或,进行 ⊕ 运算.

(12) 将第(3) 步和第(9) 步的结果相异或,进行 ⊕ 运算.

(13) 将第(2) 步和第(10) 步的结果相异或,进行 ⊕ 运算.

(14) 将第(4) 步和第(10) 步的结果相异或,进行 ⊕ 运算.

第一轮的输出是第(11)、(12)、(13)、(14) 步的结果,将中间两块((12)、(13) 的结果) 交换,作为下一轮的输入.

经过 8 轮运算之后,作最后的输出变换:

(1) x_1 与 $Z_1^{(9)}$ 子密钥进行 ⊙ 运算,结果为 y_1.

(2) x_2 与 $Z_2^{(9)}$ 子密钥进行 ⊞ 运算,结果为 y_2.

(3) x_3 与 $Z_2^{(9)}$ 子密钥进行 ⊞ 运算,结果为 y_3.

(4) x_4 与 $Z_2^{(9)}$ 子密钥进行 ⊙ 运算,结果为 y_4.

最后的密文为 $y_1y_2y_3y_4$.

子密钥为 $Z_j^{(i)}$, $i = 1,2,3,4,5,6,7,8$, $j = 1,2,3,4,5,6$, 以及 $Z_j^{(9)}$, $j = 1,2,3,4$,

共 52 个，均为 16 比特长的串.

子密钥的产生：

以 52 个子密钥在算法中的使用顺序为先后次序，IDEA 算法的主密钥 K 为 128 比特.

先将 K 分成 8 个 16 比特的子块，作为前 8 个子密钥，将 K 循环左移 25 比特得到 K'，将 K' 再分成 8 块，产生 8 个子密钥，接着将 K' 循环左移 25 比特. 如此反复，直到产生完 52 个子密钥为止.

解密过程与加密过程一致，只是使用的子密钥顺序和子密钥有所不同，解密子密钥和加密子密钥的关系及使用顺序如下：

轮数	加密子密钥	解密子密钥
1	$Z_1^{(1)}, Z_2^{(1)}, Z_3^{(1)}, Z_4^{(1)}, Z_5^{(1)}, Z_6^{(1)}$	$Z_1^{(9)-1}, -Z_2^{(9)}, -Z_3^{(9)}, Z_4^{(9)-1}, Z_5^{(8)}, Z_6^{(8)}$
2	$Z_1^{(2)}, Z_2^{(2)}, Z_3^{(2)}, Z_4^{(2)}, Z_5^{(2)}, Z_6^{(2)}$	$Z_1^{(8)-1}, -Z_2^{(8)}, -Z_3^{(8)}, Z_4^{(8)-1}, Z_5^{(7)}, Z_6^{(7)}$
3	$Z_1^{(3)}, Z_2^{(3)}, Z_3^{(3)}, Z_4^{(3)}, Z_5^{(3)}, Z_6^{(3)}$	$Z_1^{(7)-1}, -Z_2^{(7)}, -Z_3^{(7)}, Z_4^{(7)-1}, Z_5^{(6)}, Z_6^{(6)}$
4	$Z_1^{(4)}, Z_2^{(4)}, Z_3^{(4)}, Z_4^{(4)}, Z_5^{(4)}, Z_6^{(4)}$	$Z_1^{(6)-1}, -Z_2^{(6)}, -Z_3^{(6)}, Z_4^{(6)-1}, Z_5^{(5)}, Z_6^{(5)}$
5	$Z_1^{(5)}, Z_2^{(5)}, Z_3^{(5)}, Z_4^{(5)}, Z_5^{(5)}, Z_6^{(5)}$	$Z_1^{(5)-1}, -Z_2^{(5)}, -Z_3^{(5)}, Z_4^{(5)-1}, Z_5^{(4)}, Z_6^{(4)}$
6	$Z_1^{(6)}, Z_2^{(6)}, Z_3^{(6)}, Z_4^{(6)}, Z_5^{(6)}, Z_6^{(6)}$	$Z_1^{(4)-1}, -Z_2^{(4)}, -Z_3^{(4)}, Z_4^{(4)-1}, Z_5^{(3)}, Z_6^{(3)}$
7	$Z_1^{(7)}, Z_2^{(7)}, Z_3^{(7)}, Z_4^{(7)}, Z_5^{(7)}, Z_6^{(7)}$	$Z_1^{(3)-1}, -Z_2^{(3)}, -Z_3^{(3)}, Z_4^{(3)-1}, Z_5^{(2)}, Z_6^{(2)}$
8	$Z_1^{(8)}, Z_2^{(8)}, Z_3^{(8)}, Z_4^{(8)}, Z_5^{(8)}, Z_6^{(8)}$	$Z_1^{(2)-1}, -Z_2^{(2)}, -Z_3^{(2)}, Z_4^{(2)-1}, Z_5^{(1)}, Z_6^{(1)}$
输出变换	$Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)}$	$Z_1^{(1)-1}, -Z_2^{(1)}, -Z_3^{(1)}, Z_4^{(1)-1}$

其中 $Z_i^{(j)-1}$ 表示 $Z_i^{(j)}$ 的模 $\odot$ 乘法逆，$-Z_i^{(j)}$ 表示 $Z_i^{(j)}$ 的模 $\boxplus$ 加法逆.

IDEA 加密算法是近几年涌现的比较有希望的一种. 自从 DES 算法颁布以来，世界各地相继出现了多种加密算法，之所以出现这些算法，有政治原因和技术原因，各国在商用方面都需要有自己设计的加密算法，也因为 DES 算法的密钥太短，抗攻击强度不够，所以必须设计出更高强度的密码算法，以代替 DES. 这些算法有：LUCIFER 算法、Madryga 算法、NewDES 算法、FEAL-N 算法、REDOC 算法、LOKI 算法、KHUFU 算法、KHAFRE 算法、RC2 及 RC4 算法、IDEA 算法、MMB 算法、CA-1.1 算法、SKIPJACK 算法、Karn 算法以及 MDC 算法等，其中多数算法为专利算法.

以上这些算法有些已经遭到了破译，有些安全强度不如 DES，有些强度高于 DES，有些强度不明，还有待于进一步分析. 其中安全强度高于 DES 算法的有 RC2 及 RC4算法、IDEA算法、SKIPJACK算法等. 由于分组密码算法的设计与描述往往需要很长篇幅，破译与分析更要大量数学推导，在此不作详细描述.

几乎所有的分组密码体制（DES、IDEA 或其它分组密码）的工作方式都有 4 种，电子密本（ECB）、密码分组链接（CBC）、输出反馈（OFB）和密文反馈（CFB）.

（1）ECB 方式　　给定明文 x，将其分成 64 比特的明文块 $x_1, x_2, \cdots$，对每块，用

同一密钥加密得到相应的密文块,产生密文块串 $y_1, y_2, \cdots$.

(2) CBC方式 定义初始向量 $IV = y_0$,加密规则为 $y_i = E_K(y_{i-1} \oplus x_i), i > 0$.

(3) OFB方式 定义初始向量 $IV = z_0$,然后对 z_0 加密迭代,$z_i = E_K(z_{i-1}), i > 0$,产生序列流 $z_1, z_2, \cdots$,通过计算 $y_i = z_i \oplus x_i, i > 0$ 对 $x_1, x_2, \cdots$ 加密.

(4) CFB方式 定义初始向量 $IV = y_0$,产生密钥流 $z_i = E_K(y_{i-1}), i > 0$,而密文 $y_i = z_i \oplus x_i, i > 0$.

对以上四种加密方式,很容易实现对应的解密方式.

总之,因为对称密钥密码系统具有加解密速度快,安全强度高等优点,在军事、外交以及商业应用中使用越来越普遍,由于存在密钥发行与管理方面的不足,在提供数据签名、身份验证等方面需要和下面介绍的公开密钥密码系统共同使用,可以达到更好的安全效果.

3 非对称密钥密码体制

如DES、IDEA等分组密码都是对称密码,即通信双方用的密钥私下约定.这在网络上使用十分不方便.若网上有 n 个用户,两两一个密钥,则需 $\binom{n}{2}$ 个密钥,n 比较大时密钥的管理十分困难.特别是密钥的更换极为繁琐.

1976年,迪菲(Diffie)和海尔曼(Hellman)以及默考尔(Merkle)分别提出了公开密钥密码体制的思想,这不同于传统的对称密钥密码体制.它要求密钥成对出现,一个为加密密钥(e),另一个为解密密钥(d),且不可能从其中一个推导出另一个.

公钥思想的提出在近代密码发展史上是一件重大的事件.

3.1 RSA公开密钥密码算法

公开密钥:$n = pq$(p、q 分别为两个互异的大素数,p、q 必须保密),e 与 $(p-1)(q-1)$ 互素,公开(n, e).

秘密密钥:$d = e^{-1}(\mathrm{mod}(p-1)(q-1))$.

加密:$c = m^e(\mathrm{mod}\, n)$,其中 m 为明文,c 为密文.

解密:$m = c^d(\mathrm{mod}\, n)$.

安全素数:安全素数要求 p、q 满足:①$p-1, q-1$ 有大的素因子 p' 和 q';②$p'-1$ 和 $q'-1$ 也有大的素因子;③$p+1, q+1$ 有大的素因子.强素数要求 $(p-1)/2$ 和 $(q-1)/2$ 均为素数.

一般,要求使用安全素数时,n 的长度应足够大,比如十进制数200位以上.这主要是因为RSA算法的安全性依赖于对 n 的因子分解.

3.2 Rabin密码算法

设 $n = pq$ 为Blum整数,即 p、q 均为素数且 $p = 3(\mathrm{mod}4)$,$q = 3(\mathrm{mod}4)$,m 为

明文.

加密过程:$c = m^2 (\bmod n)$.

解密过程:由于收方知道 p、q,收方利用中国剩余定理计算

$$m_1 = c^{(p+1)/4} (\bmod n),$$
$$m_2 = p - c^{(p+1)/4} (\bmod n),$$
$$m_3 = c^{(q+1)/4} (\bmod n),$$
$$m_4 = q - c^{(q+1)/4} (\bmod n).$$

m_1、m_2、m_3、m_4 之一为 m. 为了正确解密,必须对 m 增加语言冗余度.

定理 1 设 $n = pq$,在模 n 下求平方根等价于对 n 因子分解.

定理 2 破译 Rabin 算法等价与对 n 因子分解.

对于RSA体制,还不能确定它的破译的难度与因子分解 n 相当. 虽然因子分解 n 可以攻破 RSA 系统.

目前无论是软件实现 RSA 还是硬件实现 RSA,速度无法同对称密钥密码算法相比. 近几年因子分解问题得到了长足发展,1995 年人类成功地分解了 129 位十进制数 RSA 密码算法.

3.3 带有加密的数字签名

像DES、IDEA等传统密码通信一样,密钥 k 是由通信双方约定,双方都掌握. 收信方可以更改密文内容,发信方可以抵赖,如若发生纠纷,无法确认究竟谁是谁非. 于是提出了"数字签名"的思想,比如 RSA 公钥便具有这样一种功能.

在公开密钥密码系统中,往往采用以下方式进行安全通信.

(1) A 用自己的私钥对信息签名 $S_A(M)$.

(2) A 用 B 的公钥加密 $E_B(S_A(M))$,发送给 B.

(3) B 用自己的私钥解密.

(4) B 用 A 的公钥验证.

另一种加密方法为

(1) A 用 K_A 对 M 加密得 c 发送给 B.

(2) B 用 K_B 对 c 加密得 c' 发送给 A.

(3) A 对 c' 解密得 c'' 发送给 B.

(4) B 对 c'' 解密.

这里没有要求 A、B 采用什么密码算法,但是要求 A 与 B 的算法具有可交换性.

3.4 Diffie-Hellman 密钥交换体制

设 p 为 512 比特以上大素数,$g < p$,p、g 公开,A 与 B 通过对称密钥密码体制进行保密通信. 以下是 A、B 通过公开密钥算法协商通信密钥的协议.

(1) A 随机选择 $x < p$,发送 $g^x(\bmod p)$ 给 B.
(2) B 随机选择 $y < p$,发送 $g^y(\bmod p)$ 给 A.
(3) A 通过自己的 x 秘密计算 $(g^y)^x(\bmod p) = g^{xy}(\bmod p)$.
(4) B 通过自己的 y 秘密计算 $(g^x)^y(\bmod p) = g^{xy}(\bmod p)$.

A 与 B 拥有相同的数据 $g^{xy}(\bmod p)$ 作为共同的秘密密钥进行保密通信.

这里的算法安全性主要依赖于有限域上的离散对数问题.

3.5　DSA 美国数字签名算法

公开密钥:p 为 512 至 1024 比特的素数(用户组公用),
q 为$(p-1)$ 的 160 比特长的素因子(用户组公用),
$g = h^{(p-1)/q}(\bmod q)$,其中 $h < p-1$,且 $g > 1$,(用户组公用),
$y = g^x(\bmod p)$.

秘密密钥:$x < q$.

签名过程:$k < q$,随机选择,
$r = (g^k(\bmod p))(\bmod q)$, $s = (k^{-1}(H(m) + xr))(\bmod q)$,
(r,s) 作为对消息 m 的签名,$H(x)$ 为安全的散列(Hash)函数.

验证过程:$w = s^{-1}(\bmod q)$,
$u1 = (H(m)w)(\bmod q)$,
$u2 = (rw)(\bmod q)$,
$v = ((g^{u1}y^{u2}(\bmod p))(\bmod q)$.

若 $v = r$,则对 m 的签名有效.其中 $H(x)$ 可选择美国推荐的标准算法 SHA 或 MD5 等安全散列算法(参见散列函数部分).

DSA 算法的安全性也依赖于有限域上的离散对数问题,安全强度和速度均低于 RSA 算法.其优点是不涉及专利问题.

3.6　ElGamal 算法

首先选择大素数 p、两个随机数 $g < p$ 和 $x < p$,计算 $y = g^x(\bmod p)$.

公开密钥是:p、g、y,秘密密钥是 x、g 和 p,在一个用户组中可以共用.

ElGamal 签名体制:

对消息 m 签名:

(1) 选择随机数 $k < p$,$(k,p-1) = 1$,计算 $a = g^k(\bmod p)$.
(2) 利用扩展的欧几里德算法计算 b,满足:

$$m = xa + kb\ (\bmod(p-1)).$$

(3) 对 m 的签名为(a,b),要求 k 保密.
(4) 如果 $y^a a^b(\bmod p) = g^m(\bmod p)$,签名有效,否则,无效.

ElGamal 加密体制:

(1) 选择随机数 $k < p$,$(k,p-1) = 1$,计算 $a = g^k(\bmod p)$.

(2) 计算 $b = y^k m \pmod p$.

(3) 密文为(a, b),其长度是明文的两倍,要求 k 保密.

(4) 解密过程为 $m = b/a^x \pmod p$.

3.7 背包公钥密码体制

将一个密码体制建立在复杂性理论中的NPC问题上,使得破译该密码系统等价于解NPC问题是密码算法设计的主要目标之一,但是目前还没有实用的这种密码系统.背包公钥密码体制是最先基于NPC问题的密码算法之一,但是由于陷门的缺陷,Merkle-Hellman背包体制及其几个变形体制都被破译了.使用的方法主要是Lenstra的整数规划方法,可参考有关资料中的 L^3 算法.

背包(子集合)问题:

设 $I = (s_1, s_2, \cdots, s_n, T)$,这里 $s_1, s_2, \cdots, s_n, T$ 是正整数, s_i 的值称为规模, T 称为目标和.

问题:存在一个0-1向量$(x_1, x_2, \cdots, x_n)$满足 $x_1 s_1 + x_2 s_2 + \cdots + x_n s_n = T$ 吗?

背包问题属于NPC问题,即没有解背包问题的多项式时间算法.

在以上的背包问题中,如果对任意的 $2 \leqslant j \leqslant n$,都有 $s_j > s_1 + s_2 + \cdots + s_{j-1}$,则称此背包问题为超递增背包问题.解超递增背包问题存在 $O(n)$ 时间的算法,且解是唯一的.

背包密码体制:

(1) 设$(s_1, s_2, \cdots, s_n)$是一超递增序列,选择大素数 $p > s_1 + s_2 + \cdots + s_n$,以及正整数 a, $0 < a < p$,

(2) 做变换: $t_i = a s_i \pmod p$, $i = 1, 2, \cdots, n$,

(3) 公开$(t_1, t_2, \cdots, t_n)$作为公开密钥.

(4) 加密:设明文消息为$(m_1, m_2, \cdots, m_n)$, $m_i = 0$或1,密文 $T = m_1 s_1 + m_2 s_2 + \cdots + m_n s_n$,

(5) 解密:计算 $Z = a^{-1} T \pmod p$,解超递增背包问题$(s_1, s_2, \cdots, s_n, Z)$,其解为明文$(m_1, m_2, \cdots, m_n)$.

3.8 McEliece密码体制

该算法主要基于一般线性码的译码问题属于NPC,而Goppa码有快速译码算法,将Goppa码伪装成一般线性码,进而设计成为一种公开密钥密码系统.所有讨论均在GF(2)域上进行.

令 $d_H(X, Y)$ 表示 X 到 Y 的汉明距离, n、k、t 是密码系统的参数.秘密密钥由三部分组成: G' 是一个能纠 t 个错误的 $k \times n$ 阶Goppa码的生成矩阵, P 是一个 $n \times n$ 置换矩阵, S 是一个 $k \times k$ 可逆矩阵;公开密钥为 $k \times n$ 矩阵 G, $G = SG'P$;明文 m 是 k 比特长的比特串.

加密过程:随机选择一个 n 维向量 z,其汉明重量小于或等于 t,

$$c = mG + z.$$

解密过程:计算 $c' = cP^{-1}$,利用 Goppa 码的译码算法对 c' 解码得 m',它满足 $d_H(m'G, c) \leqslant t$,最后计算明文 $m = m'S^{-1}$.

最初建议的参数是:$n = 1024, t = 50, k = 524$,1991 年该系统遭到了破译.

3.9 基于椭圆曲线上的公开密钥密码算法

最近有一新的研究动向,将代数几何中研究的椭圆曲线用到密码上,虽然还没有实用的例子,但值得注意.

设 p 是一个大于 3 的素数,在 GF(p) 上的椭圆曲线 E 由满足同余方程

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

的解 (x, y) 和无穷远点 O 组成.这里 a、b 是属于 GF(p) 且满足 $4a^3 + 27b^2 \neq 0 \pmod{p}$ 的常数.

定义曲线 E 上的加法"+",设 $P = (x_1, y_1)$,$Q = (x_2, y_2)$ 是 E 上的点,如果 $x_1 = x_2$,$y_1 = -y_2$,那么 $P + Q = O$,否则 $P + Q = (x_3, y_3)$,其中

$$x_3 = k^2 - x_1 - x_2,$$
$$y_3 = k(x_1 - x_3) - y_1.$$

这里,如果 $P \neq Q$,$k = (y_2 - y_1)/(x_2 - x_1)$,如果 $P = Q$,$k = (3x_1^2 + a)/2y_1$,所有的算术运算是在 GF(p) 上进行的.

E 上的所有点和 O 在上面的加法运算下构成阿贝尔群,O 为单位元.用 $\# E$ 表示曲线 E 上的点的个数.

定理 3 $p + 1 - 2p^{1/2} \leqslant \# E \leqslant p + 1 + 2p^{1/2}$.也就是说 E 上大约有 p 个点.

定理 4 设 E 是 GF(p) 上的椭圆曲线,p 是大于 3 的素数,那么存在正整数 n、n',使得 E 同构于 $Z_n \times Z_{n'}$,而 $n' \mid n$,且 $n' \mid (p - 1)$.

如果 $n' = 1$,那么 E 是一个循环群,如果能够计算出 n、n',那么可以在 E 或其子群上实现 ElGamal 密码系统.

椭圆曲线 E 上的离散对数问题:设 P_0 属于 E,为 E 的循环子群 H 的生成元,任给正整数 k,计算 kP_0 是容易的,而任给 Q 属于 H,计算 k 使得 $kP_0 = Q$,称为椭圆曲线 E 上的离散对数问题.解此问题是困难的.

椭圆曲线上的 ElGamal 密码系统:

(1) P_0 属于 E,任选正整数 x,$Y = xP_0$,公开 Y、P_0、x 作为秘密密钥.

(2) 加密:随机选择正整数 k,计算 $C = kP_0$、设明文 m 为 E 上的点,那么,密文为 $(C, m + kY)$.

(3) 解密:计算点 $Q = xC = kY$,$m = m + kY - Q$.

这里要求将明文嵌入到 E 上,可以有多种编码方式.下面的 Menezes-Vanstone 体制避免了对明文的嵌入.

(1) 设 P_0 属于 E,任选正整数 x,$Y = xP_0$,公开 Y、P_0、x 作为秘密密钥.

(2) 加密:随机选择正整数 k,计算 $C = kP_0$,设明文 $m = (m_1, m_2)$,那么,密文

为(C, y_1, y_2)，记$(c_1, c_2) = kY$，其中 $y_1 = c_1 m_2 (\bmod p)$，$y_2 = c_2 m_2 (\bmod p)$.

(3) 解密：计算$(y_1 c_1^{-1} (\bmod p), y_2 c_2^{-1} (\bmod p)) = m$，其中 $xC = (c_1, c_2)$.

3.10　ESIGN 算法

该算法是用来进行数字签名的. 秘密密钥是一对大素数 p、q，公开密钥是 $n = p^2 q$，m 是要签名的消息. $H(x)$ 是一安全的散列函数，k 为一安全参数，它们是公开的.

签名过程为：

(1) A 选择随机数 $x < pq$.

(2) A 计算 w，w 是大于等于$(H(m) - x^k (\bmod n))/pq$ 的最小整数.

(3) A 计算 $s = x + ((w/kx^{k-1})(\bmod p))pq$.

(4) A 发送 m 及其签名 s 给 B.

验证过程为：

(1) B 计算 $t = s^k (\bmod n)$.

(2) B 计算 a，a 是大于或等于 n 的比特数的 2 倍除以 3 的最小整数.

(3) 如果 $H(m) \leq t < H(m) + 2^a$，那么该签名有效，否则无效.

在几乎所有的实用公开密钥密码系统中，都涉及到大数运算和素数选择. 模幂运算采用反复平方取模算法，素数测试一般采用 Rabin-Miller 算法. 还有其它素性测试算法用来选择大素数，如 Solovag-Strassen 测试法，Lehmann 测试法，等等.

Rabin-Miller 算法：

设待测素数为 p，$p = 1 + 2^b m$

(1) 随机选择正整数 $a < p$.

(2) 计算 $z = a^m (\bmod p)$.

(3) 如果 $z = 1$，那么回答 p 是素数，退出.

(4) 对于 $i = 0$ 至$(b - 1)$，如果 $z = p - 1$，那么回答 p 是素数，退出，

否则计算 $z = z^2 (\bmod p)$，回答 p 不是素数，结束.

如果算法结束，回答 p 是素数，就算 p 通过一次测试. 通过一次测试，产生一个假素数的概率小于 1/4，对一个数连续测试 n 次，都通过了测试，那么此数不是素数的概率小于 $1/4^n$.

4　散 列 函 数

在网络上传递的信息很多情况不一定需要保密，但要求对它的完整性进行确认，即确定它的确是由发信方送来的，没被修改. 数字签名便是其中一种. 散列(hash) 函数源于密码，用途很广. 数字签名技术中也经常用到单向散列函数 $H(x)$，也称杂凑函数或消息(报文)摘要，其作用是将任意长度的消息 m 压缩成一固定长

度的散列值 h，$h = H(m)$. 如果将不可更改的 h 附在消息 m 之后或对 h 签名，可以防止对消息 m 的篡改、抵赖或伪造.

通常 $H(x)$ 必须满足以下性质：

(1) 快速性：对任意长度的 m，计算 $h = H(m)$ 很容易，多项式时间可计算.

(2) 单向性：给定 h，计算 m，使得 $h = H(m)$ 很困难.

(3) 无碰撞性：给定 m，要找到另一消息 m'，满足 $H(m') = H(m)$ 很困难，在计算上不可行，称 H 是弱无碰撞的；如果找到 $m \neq m'$，满足 $H(m') = H(m)$，在计算上是不可行的，称 H 是强无碰撞的.

强无碰撞性包含弱无碰撞性.

生日攻击：

定理 1　随机地将 k 个球抛入 n 个盒子，若某个盒子至少含有两个球的概率为 α，则 $k \approx (2n\ln(1/(1-\alpha)))^{1/2}$.

如 $\alpha = 0.5$，则 $k \approx 1.17n^{1/2}$，即当 $H(x)$ 的长度为 n 比特时，大约在 $2^{n/2}$ 个消息中有两个相同的消息摘要的概率为 50%.

4.1　SHA 算法

SHA 算法在 DSA 标准中使用.

初始变量：A = 67452301，B = EFCDAB89，C = 98BADCFE，D = 10325476，E = C3D2E1F0，K_1 = 5A827999，K_2 = 6ED9EBA1，K_3 = 8F1BBCDC，K_4 = CA62C1D1，

四个迭代函数：$f_1(X,Y,Z) = XY\text{or}(\text{not}X)Z$，　$f_2(X,Y,Z) = XX\text{or}YX\ \text{or}Z$，

$f_3(X,Y,Z) = XY\text{or}XZ\text{or}YZ$，　$f_4(X,Y,Z) = XX\text{or}YX\ \text{or}Z$，

设消息为 512 比特的 m_i，$i = 0,1,\cdots,15$，$W_i = m_i$，$l = 0,1,\cdots,15$，

$$W_i = W_{i-3}X\text{or}\ W_{i-8}X\text{or}\ W_{i-14}X\text{or}\ W_{i-16}，\quad l = 16,17,\cdots,79,$$

按以下步骤迭代 80 圈：

$$\text{temp} = (A <<< 5) + f_i(B,C,D) + E + W_j + K_i,$$

$E = D$，

$D = C$，

$C = (B <<< 30)$，

$B = A$，

$A = \text{temp}$，

$i = 1,2,3,4$，$j = 0,1,\cdots 79$，对每个 i，各迭代 20 圈，最后输出的 ABCDE 与初始值 ABCDE 异或，得到最终结果 ABCDE 共 160 比特. 上面的"+"为异或运算，"<<<"为循环左移.

4.2　MD5 算法

MD5 算法的输入为 512 比特，散列输出为 128 比特.

(1) 4 个 32 比特的初始变量：

A = AA = 01234567,　　B = BB = 89ABCDEF.
C = CC = FEDCBA98,　　D = DD = 76543210.

(2) 4个非线性函数:

$F(X,Y,Z) = XY\text{or}(\text{not}X)Z$,　　$G(X,Y,Z) = XZ\text{or}Y(\text{not }Z)$,
$H(X,Y,Z) = XX\text{or}YX\text{or}Z$,　　$I(X,Y,Z) = YX\text{or}(X\text{or}(\text{not }Z))$.

(3) 4种操作:

Round($x,a,b,c,d,M[j],s,t_i$) 表示 $a = b + (a + x(b,c,d) + M[j] + t_i) <<< s)$,

x 表示上面4种函数 F,G,H,I 之一. $M[j], j = 0,1,\cdots,15$ 表示16个32比特明文. t_i 为不同的常数, s 为循环移位量.

(4) 按以下操作4轮,每轮16个操作.

第一轮:

Round(F,A,B,C,D,X[0],7,0xd76aa478),
Round(F,D,A,B,C,X[1],12,0xe8c7b756),
Round(F,C,D,A,B,X[2],17,0x242070db),
Round(F,B,C,D,A,X[3],22,0xc1bdceee),
Round(F,A,B,C,D,X[4],7,0xf57c0faf),
Round(F,D,A,B,C,X[5],12,0x4787c62a),
Round(F,C,D,A,B,X[6],17,0xa8304613),
Round(F,B,C,D,A,X[7],22,0xfd469501),
Round(F,A,B,C,D,X[8],7,0x698098d8),
Round(F,D,A,B,C,X[9],12,0x8b44f7af),
Round(F,C,D,A,B,X[10],17,0xffff5bb1),
Round(F,B,C,D,A,X[11],22,0x895cd7be),
Round(F,A,B,C,D,X[12],7,0x6b901122),
Round(F,D,A,B,C,X[13],12,0xfd987193),
Round(F,C,D,A,B,X[14],17,0xa679438e),
Round(F,B,C,D,A,X[15],22,0x49b40821).

第二轮:

Round(G,A,B,C,D,X[1],5,0xf61e2562),
Round(G,D,A,B,C,X[6],9,0xc040b340),
Round(G,C,D,A,B,X[11],14,0x265e5a51),
Round(G,B,C,D,A,X[0],20,0xe9b6c7aa),
Round(G,A,B,C,D,X[5],5,0xd62f105d),
Round(G,D,A,B,C,X[10],9,0x02441453),
Round(G,C,D,A,B,X[15],14,0xd8a1e681),
Round(G,B,C,D,A,X[4],20,0xe7d3fbc8),
Round(G,A,B,C,D,X[9],5,0x21e1cde6),
Round(G,D,A,B,C,X[14],9,0xc33707d6),

Round(G,C,D,A,B,X[3],14,0xf4d50d87),
Round(G,B,C,D,A,X[8],20,0x455a14ed),
Round(G,A,B,C,D,X[13],5,0xa9e3e905),
Round(G,D,A,B,C,X[2],9,0xfcefa3f8),
Round(G,C,D,A,B,X[7],14,0x676f02d9),
Round(G,B,C,D,A,X[12],20,0x8d2a4c8a).

第三轮:

Round(H,A,B,C,D,X[5],4,0xfffa3942),
Round(H,D,A,B,C,X[8],11,0x8771f681),
Round(H,C,D,A,B,X[11],16,0x6d9d6122),
Round(H,B,C,D,A,X[14],23,0xfde5380c),
Round(H,A,B,C,D,X[1],4,0xa4beea44),
Round(H,D,A,B,C,X[4],11,0x4bdecfa9),
Round(H,C,D,A,B,X[7],16,0xf6bb4b60),
Round(H,B,C,D,A,X[10],23,0xbebfbc70),
Round(H,A,B,C,D,X[13],4,0x289b7ec6),
Round(H,D,A,B,C,X[0],11,0xeaa127fa),
Round(H,C,D,A,B,X[3],16,0xd4ef3085),
Round(H,B,C,D,A,X[6],23,0x04881d05),
Round(H,A,B,C,D,X[9],4,0xd9d4d039),
Round(H,D,A,B,C,X[12],11,0xe6db99e5),
Round(H,C,D,A,B,X[15],16,0x1fa27cf8),
Round(H,B,C,D,A,X[2],23,0xc4ac5665).

第四轮:

Round(I,A,B,C,D,X[0],6,0xf4292244),
Round(I,D,A,B,C,X[7],10,0x432aff97),
Round(I,C,D,A,B,X[14],15,0xab9423a7),
Round(I,B,C,D,A,X[5],21,0xfc93a039),
Round(I,A,B,C,D,X[12],6,0x655b59c3),
Round(I,D,A,B,C,X[3],10,0x8f0ccc92),
Round(I,C,D,A,B,X[10],15,0xffeff47d),
Round(I,B,C,D,A,X[1],21,0x85845dd1),
Round(I,A,B,C,D,X[8],6,0x6fa87e4f),
Round(I,D,A,B,C,X[15],10,0xfe2ce6e0),
Round(I,C,D,A,B,X[6],15,0xa3014314),
Round(I,B,C,D,A,X[13],21,0x4e0811a1),
Round(I,A,B,C,D,X[4],6,0xf7537e82),
Round(I,D,A,B,C,X[11],10,0xbd3af235),
Round(I,C,D,A,B,X[2],15,0x2ad7d2bb),

Round($I,B,C,D,A,X[9],21,0xeb86d391$).

所有操作进行完了以后，将 A、B、C、D 与 AA、BB、CC、DD 异或，然后代入下一512比特进行计算，最后输出为 A、B、C、D 的级联，共128比特.

4.3　基于分组密码算法的散列函数

设一个安全的分组密码算法为 $c = f(m,k)$，对任意的明文 $x = x_1x_2\cdots x_n$，其散列值的计算过程如下：

(1) 给定初值 $g_0 = IV$.

(2) 通过迭代 $g_i = f(x_i, g_{i-1})$.

(3) 最后 $h(x) = g_n$.

这里的 f 可以是DES、IDEA等，主要是对算法的输入、输出、密钥长度做适当调整.

另外4种迭代模式是：

$$g_i = f(x_i, g_{i-1}) \oplus x_i,$$
$$g_i = f(x_i, g_{i-1}) \oplus x_i \oplus g_{i-1},$$
$$g_i = f(x_i \oplus g_{i-1}, g_{i-1}) \oplus x_i,$$
$$g_i = f(x_i \oplus g_{i-1}, g_{i-1}) \oplus x_i \oplus g_{i-1}.$$

4.4　基于离散对数的散列函数

设 p 是一个大素数，$q = (p-1)/2$ 也是素数，α、β 是 GF(p) 上的两个本原元，$\log_\alpha\beta$ 不公开，且假设计算它是不可能的，散列函数 h 为将 $\{0,1,2,\cdots,q-1\} \times \{0,1,2,\cdots,q-1\}$ 映射到 GF(p) \ $\{0\}$ 上的函数，形式如下：$h(x,y) = \alpha^x\beta^y \pmod p$，$0 \leqslant x, y \leqslant q-1$.

定理2　给定以上散列函数的一个碰撞，那么离散对数 $\log_\alpha\beta$ 就能计算.

4.5　扩展的散列函数

以上的散列函数大都是基于固定长度输入、固定长度输出的函数，下面几种方法可以将一个满足强碰撞条件的散列函数扩展成另外一个满足强碰撞条件的散列函数.

设 h 是 GF(2^m) 到 GF(2^t) 的满足强碰撞条件的散列函数，串 x 的长度 $|x| = n > m$，$x \| y$ 表示 x 后面级联 y. 设 $x = x_1 \| x_2 \| \cdots \| x_k$，$|x_1| = |x_2| = \cdots = |x_{k-1}| = m-t-1$，$|x_k| = m-t-1-d$，$k = \lceil n/(m-t-1) \rceil$. 记 $y(x) = y_1 \| y_2 \| \cdots \| y_k \| y_{k+1}$，$y_i = x_i$，$i = 1,2,\cdots,k-1$，$y_k = x_k \| 0^d$，即 x_k 后补 d 个0，y_{k+1} 为 d 的二进制表示. 当 $m > t+1$ 时，扩展过程为：

(1) 设 $g_1 = h(0^{t+1} \| y_1)$，

(2) 对 $i = 1$ 到 k 做 $g_{i+1} = h(g_i \| 1 \| y_{i+1})$，

(3) $h^*(x) = g_{k+1}$.

定理 3 h^* 也是强无碰撞的散列函数.

当 $m = t+1$ 时,使用以下扩展方法构造的散列函数 h^* 也是强无碰撞的.设函数 f 为 $f(0) = 0, f(1) = 01, x = x_1x_2\cdots x_n, x_i = 0$ 或 1.

(1) 设 $y = y_1y_2\cdots y_k = 11 \| f(x_1) \| f(x_2) \| \cdots \| f(x_n)$，

(2) $g_1 = h(0^t \| y_1)$，

(3) 对 $i = 1$ 到 $k-1$ 做 $g_{i+1} = h(g_i \| y_{i+1})$，

(4) $h^*(x) = g_k$.

5 秘密分存

有些重要的秘密,比如极为重要的密钥,由一个人掌握不安全,需要若干人同时到场时方可决定.所以有秘密分存或密钥分存问题.把一个秘密 m 划分成任意的 n 片分给任意 n 个人,每片称为一个影子,使得他们当中任意 k 个影子合起来,可恢复出秘密 m,少于 k 个影子在一起得不到 m 的任何信息.称这样的方案为 (k,n) 门限方案(或分存方案),其中 $k < n$.

沙米尔(A.Shamir)最先给出了基于多项式插值的解决方案,后来又有 Blakley 方案、Asmuth-Bloom 方案等.

5.1 沙米尔多项式插值法

设 p 是一素数,p 大于 n 和 m,产生一个 $k-1$ 次多项式:

$$f(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + a_1x + a_0 \pmod{p},$$

其中 $a_0 = m, a_i < p$ 是随机选择的.

计算影子:选择 n 个不同的点 $x_i, k_i = f(x_i), i = 1,2,\cdots,n$,分给第 i 个人的影子为 $(x_i, k_i), i = 1,2,\cdots,n$.

当有多于 k 个影子在一起时,通过解线性方程组,恢复出 $f(x)$ 的所有系数,进而知道 m,而少于 k 个影子就不能.

例 1 设 $m = 11$,构造(3,5)门限方案.选择 $p = 13$,产生 2 次方程

$$f(x) = 7x^2 + 8x + 11 \pmod{13},$$

其中 8、7 是随机选择的.取 $x_i = i, i = 1,2,3,4,5$,那么

$$k_1 = f(1) = 0 \pmod{13},$$
$$k_2 = f(2) = 3 \pmod{13},$$
$$k_3 = f(3) = 7 \pmod{13},$$
$$k_4 = f(4) = 12 \pmod{13},$$
$$k_5 = f(5) = 5 \pmod{13}.$$

例如用 k_2、k_3、k_5 来恢复 m，代入 $f(x)$，解以下线性方程组：

$$3 = 2^2a + 2b + m \pmod{13},$$
$$7 = 3^2a + 3b + m \pmod{13},$$
$$5 = 5^2a + 5b + m \pmod{13},$$

其解为 $a = 7, b = 8, m = 11$.

5.2　其它门限方案

1. 矢量方案(Blakley 方案)

秘密 m 被定义为 k 维空间的一个点，每个影子是包含这个点的 $(k-1)$ 维超平面(方程). 任意 k 个这样的超平面相交的唯一点刚好是这个点.

2. 中国剩余定理方案(Asmuth-Bloom 方案)

选择一个素数 p，$p > m$，然后选择 n 个比 p 小的数 $d_1, d_2, \cdots, d_n$，使它们按递增顺序排列：$d_i < d_{i+1}$，对任意的 i、$j(d_i, d_j) = 1$，并且满足条件

$$d_1 d_2 \cdots d_k > p d_{n-k+2} d_{n-k+3} \cdots d_n.$$

(1) 选择随机数 r.

(2) 计算 $m' = m + rp$.

(3) 影子 $k_i = m' \pmod{d_i}$.

利用中国剩余定理，任意 k 个影子就可恢复 m，而少于 k 个不行.

3. 矩阵方案(Kamin-Greene-Hellman 方案)

选择 $n+1$ 个 k 维列矢量 $V_0, V_1, \cdots, V_n$，使得由它们形成的任意 $k \times k$ 阶矩阵的秩为 k，向量 U 是 k 维行向量，秘密信息为 $m = U \cdot V_0$，影子 $k_i = U \cdot V_i$，$i = 1, 2, \cdots, n$.

恢复秘密 m 可由任意 k 个影子解含 k 个未知数的线性方程组求出 U，再由 U 恢复出 m.

根据以上的门限方案，可以设计更加复杂的门限方案. 例如：

(1) 一个人比其它人更重要，需要给他更多的影子.

(2) 两个代表团 A、B，只有 A 中的 k 个人和 B 中的 j 个人参加才能恢复出秘密 m，A 与 B 是敌对的，互不信任.

(3) 在大家聚在一起恢复秘密时，不泄漏各自持有物.

(4) 在有骗子情况下，如果有人进行欺骗，便不能恢复正确秘密；设计一种能够发现骗子的方案.

(5) 有预防情况的共享方案，方案要求只有 n 个人中的某些人在一起时能恢复秘密，而其它人再多也不行.

6　安全协议与密钥管理

信息安全除了密码算法以外，还包括如何安全地使用这些算法，安全地管理信

息系统等.安全管理不仅包括用户的访问控制,更重要的是密钥的产生、分发、使用等关键环节.例如要求密钥长度128比特,这128比特必须是绝对随机产生的,稍有不慎,可能造成安全隐患.

安全协议是信息安全技术的另一重要课题,在允许用户违背协议的情况下,完备的安全协议可以阻止蓄意破坏者对信息系统的恶意攻击.

6.1 伪随机数生成

在序列密码中,伪随机数的产生是非常实用的方法,但是在理论上无法保证这些序列是安全的.下面是一些基于数论难题的伪随机数生成器.

1. RSA 生成器

设 p、q 是两个大素数,$n = pq$,b 满足 $(b,\phi(n)) = 1$.种子是 $s_0 < n$. n、b 公开而 p、q 保密.

对 $i = 0,1,2,\cdots$,迭代 $s_{i+1} = s_i^b \pmod n$.随机序列为 $(z_1,z_2,\cdots,z_k)$,其中 $z_i = s_i \bmod 2$.

2. Blum-Blum-Shub 生成器

设 p、q 是两个大素数,且 $p = q \pmod 4 = 3$,$n = pq$. n 公开而 p、q 保密.

$\mathrm{QR}(n)$ 表示模 n 的平方剩余的集合.

所谓 x 是模 n 的平方剩余,是指方程 $y^2 = x \pmod n$ 有解,若无解则称 x 是模 n 的非平方剩余.所有的模 n 的非平方剩余构成的集合用 $\mathrm{NQR}(n)$ 表示.

设种子 s_0 是属于 $\mathrm{QR}(n)$ 的一任意元素,对 $i = 0,1,\cdots$,迭代

$$s_{i+1} = s_i^2 \pmod n.$$

随机序列为 $(z_1,z_2,\cdots,z_k)$,其中 $z_i = s_i \pmod 2$.称随机序列 $(z_1,z_2,\cdots,z_k)$ 为 BBS 序列.

3. 离散对数生成器

设 p 是一个大素数,a 是 $\mathrm{GF}(p)$ 上的本原元.种子 $x_0 < p$,定义

$$x_{i+1} = a_i^x \pmod p,$$

随机序列为 $(z_1,z_2,\cdots,z_k)$,其中 $z_i = 1$,如果 $x_i > p/2$,否则为 0.

6.2 概率加密

1. Goldwasser-Micali 体制

设 $n = pq$,p、q 为大素数,m 属于 $NQR(n)$,n、m 公开,p、q 保密.加密信息为 $x = 0$ 或 1,选择随机数 $r < n$.

加密过程为:密文 $y = m^x r^2 \pmod n$,x 为明文.

解密:如果 y 属于 $QR(n)$,则 $x = 0$,否则 $x = 1$.

此密码体制是逐比特加密,密文扩展量巨大.

2. Blum-Goldwasser 体制

设 n 是 Blum 数，$n = pq$，n 公开，p、q 保密.

(1) 使用 BBS 发生器产生以 $s_0 = r$ 为种子的 BBS 序列$(z_1, z_2, \cdots, z_k)$.

(2) 计算出 $s_{k+1} = s_k^2 \pmod n$.

(3) 对明文0-1串 $x = x_1, x_2, \cdots, x_k$ 加密：$y_i = (x_i + z_i) \pmod 2$，$i = 1, 2, \cdots, k$.

(4) 密文为$(y_1, y_2, \cdots, y_k, s_{k+1})$.

解密：

(5) 计算 $a = ((p+1)/4)^{k+1} (\bmod (p-1))$.

(6) 计算 $b = ((q+1)/4)^{k+1} (\bmod (q-1))$.

(7) 计算 $c_1 = {s_{k+1}}^a \pmod p$.

(8) 计算 $c_2 = {s_{k+1}}^b \pmod q$.

(9) 利用中国剩余定理找到 s_0 满足：

$$s_0 = c_1 \pmod p \text{ 和 } s_0 = c_2 \pmod q.$$

(10) 以 s_0 为种子产生 BBS 序列$(z_1, z_2, \cdots, z_k)$.

(11) 明文$(x_1, x_2, \cdots, x_k) = (y_1, y_2, \cdots, y_k) \oplus (z_1, z_2, \cdots, z_k)$.

6.3　不可否认签名

不可否认签名不同于一般数字签名.普通数字签名是 A 对信息 m 的签名公开后，所有知道 A 的公开密钥的人都可对此签名验证；而不可否认签名要求对 A 签名的验证时 A 须参与，当 B 持有 A 的签名时，无法向第三者 C 证明此签名的正确性.

1. 简单的不可否认签名

(1) 选择大素数 p，设 g 是其生成元，p、g 公开. A 的秘密密钥为 x，公开密钥为 $y = g^x (\bmod\ p)$，m 为消息.

(2) A 计算 $z = m^x (\bmod\ p)$，发送 z 给 B.

(3) B 随机选择两个小于 p 的数 a、b，计算 $c = z^a y^b \pmod p$，发送 c 给 A.

(4) A 计算 $w = x^{-1} (\bmod (p-1))$，发送 $d = c^w \pmod p$ 给 B.

(5) B 验证 $d = m^a g^b \pmod p$ 是否成立，如果成立，签名有效，否则无效.

2. 复杂的不可否认签名

(1) 选择大素数 p，设 g 是其生成元，p、g 公开，A 的秘密密钥为 x，公开密钥为 $y = g^x (\bmod\ p)$，m 为消息.

(2) A 计算 $z = m^x \pmod p$，发送 z 给 B.

(3) B 随机选择两个小于 p 的数 a、b，计算 $c = m^a g^b \pmod p$，发送 c 给 A.

(4) A 选择小于 p 的随机数 q，计算 $s_1 = cg^q \pmod p$，$s_2 = (cg^q)^x \pmod p$，发送 s_1、s_2 给 B.

(5) B 把 a、b 发给 A，以证明在第(3)步没有欺骗 A.

(6) A 将 q 发给 B.

(7) B 验证，$s_1 = (cg^q) \pmod p$，$s_2 = (y^{b+q} z^a) \pmod p$ 是否成立，如果成立，签名有效，否则无效.

3. 盲签名

A 需要对信息 m 让 B 签名，而不让 B 知道 m.

设 B 有使用 RSA 算法的公开密钥 e、n，和秘密密钥 d.

(1) A 随机选择 $k < n$，隐蔽 m，$t = (mk^e)\ (\mathrm{mod}\, n)$.

(2) A 让 B 对 t 签名，$t^d = (mk^e)^d (\mathrm{mod}\, n)$.

(3) A 获得签名后，恢复对 m 的签名，$m^d = t^d / k\ (\mathrm{mod}\, n)$.

6.4　安全计算与公正掷币协议

1. 安全计算协议

设 p 是一个大素数，g 是 GF(p) 的乘法群的生成元. A 有一 $x < p$，A 想知道 e，使得 $g^e = x\ (\mathrm{mod}\, p)$，$A$ 无能力计算离散对数问题，而 B 有大的计算能力计算此问题. 如何使得在 A 不暴露 x 的情况下，让 B 帮助 A 计算此离散对数问题？协议如下：

(1) A 选择随机数 $r < p$.

(2) A 计算 $x' = xg^r (\mathrm{mod}\, p)$.

(3) A 要求 B 解 $g^{e'} = x'\ (\mathrm{mod}\, p)$.

(4) B 计算 e'，将 e' 给 A.

(5) A 计算 $e = (e' - r)\ (\mathrm{mod}(p-1))$ 来恢复 e.

对二次剩余问题和本原根问题也有类似协议.

安全的多方计算：

设 A 知道整数 i，B 知道整数 j，在不把自己的数告诉对方的情况下，判断 $i \leq j$ 或 $i > j$. 协议如下：

例如 i、j 的取值范围为 1 到 100，B 有一公开密钥和秘密密钥，E_B、D_B 分别表示用 B 的密钥进行的加密和解密运算.

(1) A 选择一个大随机数 x，用 B 的公钥加密，$c = E_B(x)$.

(2) A 计算 $c - i$，并将 $c - i$ 和 x 的规模发给 B.

(3) B 计算下面 100 个数，$y_u = D_B(c - i + u)$，$u = 1, 2, \cdots, 100$.

B 随机选择大的素数 $p < x$，(A 可以事先告诉 B 关于 x 的规模)，计算 $z_u = y_u (\mathrm{mod}\, p)$，$u = 1, 2, \cdots, 100$.

B 验证对所有的 $u \neq v$，$|z_u - z_v| \geq 2$，以及对所有的 u，$0 < z_u < p - 1$. 如果验证失败，B 重新选择另一素数 p 再试.

(4) B 把下面的一串数发给 A：

$$z_1, z_2, \cdots, z_j, z_{j+1} + 1,\quad z_{j+2} + 1, \cdots, z_{100} + 1, p.$$

(5) A 判断串中第 i 个数是否和 $x\ (\mathrm{mod}\, p)$ 同余，如果同余，其结论是 $i \leq j$，不同余，$i > j$.

(6) A 将结果告诉 B.

此协议在第(6) 步有 A 提前结束协议和对 B 欺骗的可能.

2. 公平掷币协议

基于因子分解问题的掷币协议：

(1) A 选择两个大素数 p、q，$n = pq$，将 n 发给 B.

(2) B 随机选择正整数 $r \leq n/2$,计算 $z = r^2 (\bmod n)$,将 z 发送给 A.

(3) A 计算 z 模 n 的 4 个平方根,x、$-x$、y、$-y$,设 $x' = \min(x, -x)$,$y' = \min(y, -y)$,则 r 等于 x' 或 y'.

(4) A 对 B 猜测 $r = x'$ 或 $r = y'$(方法是找到 x'、y' 的不同比特的位置 i,告诉 B:你的 r 的第 i 比特是 0,或你的 r 的第 i 比特是 1),

(5) 如果 A 的猜测正确,掷币结果为正面,否则为反面,B 宣布掷币结果.

验证子协议:

(6) B 把 r 发送给 A.

(7) A 把 p、q 发送给 B.

基于离散对数问题的掷币协议:

(1) A、B 共同选择素数 p,且知道 $p-1$ 的因子.

(2) B 选择 $GF(p)$ 的两个本原元 h、t,将它们发送给 A.

(3) A 选择随机数 x,$(x,(p-1)) = 1$,计算下面两个式子中的一个:

$$y = h^x (\bmod p) \quad 或 \quad y = t^x (\bmod p).$$

将 y 发送给 B.

(4) B 猜测 y 是 h 的函数还是 t 的函数,将猜测结果给 A.

(5) 如果 B 猜测正确,抛币结果为正面,否则为反面.A 宣布抛掷结果.

验证子协议:

(6) A 将 x 给 B,B 可以验证猜测结果以及 A 是否正确执行协议.

6.5 阈下信道

在数字签名系统中,A 通过对无害信息 m' 的签名给 B,将真正隐蔽的信息 m 传送给 B,m 称为阈下信息.

ElGamal 签名体制的阈下信道:

选择素数 p、随机数 $g < p$、$r < p$,计算 $K = g^r (\bmod p)$.A 的公开密钥为 K、p、g,秘密密钥为 r,B 知道 r,用来传送阈下信息 m.

设 $(m,p) = 1$,$(m',p) = 1$,$(m,p-1) = 1$.

(1) A 计算 $X = g^m (\bmod p)$.

(2) A 用欧几里德算法计算 $m' = rX + mY\ (\bmod (p-1))$.

(3) A 对消息 m' 的 ElGamal 签名为 (X,Y).

(4) 任何人可以验证 A 的 ElGamal 签名:

$$K^X X^Y (\bmod p) = g^{m'} (\bmod p).$$

(5) B 首先验证 $(g^r)^X X^Y (\bmod p) = g^{m'} (\bmod p)$,确信签名来自 A.

(6) B 恢复 $m = Y^{-1}(m' - rX)\ (\bmod (p-1))$.

此协议有 B 假冒 A 签名的风险.对许多数字签名体制,都存在阈下信道,如 DSS.

6.6 零知识证明协议

零知识证明协议要求证明者 P 向验证者 V 证明一个结论是正确的.如果 P 的结论正确,V 以接近1的概率接受,如果 P 欺骗 V,V 以接近1的概率发现 P 的欺骗行为;无论 V 是否违背协议,V 除了接受 P 的结论以外,从 P 那里得不到其它额外信息.零知识证明技术可以用来设计身份验证方案和数字签名方案.

1. Feige-Fiat-Shamir 方案

设 $n = pq$,p、q 为两个互异的大素数,P 秘密选择 s,P 计算 $v = s^2(\bmod n)$,P 公开 v、n,证明者 P 向验证者 V 证明他知道 v 的一个平方根.

(1) P 随机选择一正整数 $r < n$,计算 $x = r^2(\bmod n)$,发送 x 给 V.

(2) V 发送一个比特 $b = 0$ 或 1 给 P.

(3) P 发送 $y = rs^b(\bmod n)$ 给 V.

(4) V 验证 $y^2 = xv^b(\bmod n)$ 是否成立,若成立则继续此协议.

以上协议进行 n 圈,P 欺骗成功的概率为 $1/2^n$.

并行协议:

设 $n = pq$,p、q 为两个互异的大素数,P 秘密选择 s_1、s_2,…,s_k,P 计算 $v_i = s_i^2(\bmod n)$,P 公开 v_1,v_2,…,v_k 作为 P 的身份.

(1) P 随机选择一正整数 $r < n$,计算 $x = r^2(\bmod n)$,发送 x 给 V.

(2) V 发送一个比特串 $b = b_1b_2\cdots b_k$,(b_i 为 0 或 1) 给 P.

(3) P 发送 $y = r\prod s_i^{b_i}(\bmod n)$ 给 V.

(4) V 验证 $y^2 = x\prod v_i^{b_i}(\bmod n)$ 是否成立,若成立则 P 欺骗的概率小于 $1/2^k$.

2. Schnorr 鉴别协议

选择大素数 p、q,q 是 $p-1$ 的因子,选择 $a \neq 1$,满足 $a^q = 1\ (\bmod p)$,这些数据全部对外公开.P 随机选择 $s < q$ 作为秘密密钥,$v = a^{-s}(\bmod p)$ 作为 P 的公开密钥.

(1) P 选择随机数 $r < q$,计算 $x = a^r(\bmod p)$ 发送 x 给 V.

(2) V 发送随机数 e 给 P,$0 < e < 2^t$,t 为安全参数.

(3) P 计算 $y = r + se\ (\bmod q)$,发送 y 给 V.

(4) V 验证 $x = (a^y v^e)(\bmod p)$ 是否成立,若成立 V 则相信与之通信的是 P.

Schnorr 建议 p 为 512 比特,q 为 140 比特,$t = 72$,破译此协议的难度大约为 2^t.

3. 离散对数零知识证明协议

设 p 是大素数,g、$y = g^x(\bmod p)$ 对外公开,x 保密,P 向 V 证明他知道 x.

(1) P 选择随机数 $r < p-1$,计算 $h = g^r(\bmod p)$,发送 h 给 V.

(2) V 发送一个比特 b 给 P.

(3) P 计算 $s = r + bx\ (\bmod\ (p-1))$,发送 s 给 V.

(4) V 验证 $g^s = hy^b(\bmod p)$.

重复执行 n 次,P 欺骗成功的概率为 $1/2^n$.

4. 破译RSA的能力的零知识证明协议

设 C 的RSA的公开密钥为 e,私钥为 d,模为 n,A 破译了 C 的密码,知道了 d,A 向 B 零知识地证明这一点.

(1) A 与 B 商定随机数 k、m,使得 $km = e\ (\mathrm{mod}(p-1))$,$k > 3$,$m > 3$.

(2) A 与 B 随机产生一个密文 c(这些随机数可采用掷币协议产生).

(3) A 使用 C 的秘密密钥计算 $M = c^d(\mathrm{mod}\,n)$,然后计算 $x = M^k(\mathrm{mod}\,n)$,发送 x 给 B.

(4) B 验证 $x^m(\mathrm{mod}\,n) = c$ 是否成立,如果成立,相信 A 所说的是真的,否则,A 在欺骗.

也有类似的破译离散对数问题的零知识证明协议.在零知识证明协议中集中使用了下面的比特托管技术和遗忘传递技术,它们是设计零知识证明协议的基础.

5. 比特托管(bit commitment)

所谓的比特托管是这样一种技术,它要求 A 向 B 提交一个比特 b,B 不能打开 b,A 不能改变 b;当需要打开时,A 向 B 打开 b.掷币协议是一个很好的例子.

实现比特托管可以采用上面的掷币协议和概率加密协议,还可采用下面的量子密码技术.下面是利用比特掩盖(B.C)技术实现无向图 G 的哈密顿(Hamilton)回路的零知识证明协议.

前提:证明者 P 知道无向图 G 的一条哈密顿回路,P 向验证者 V 零知识地证明这一点,使得 V 相信图 G 是一个哈密顿图,而不告诉 V 这条哈密顿回路.

(1) P 随机的选择一个置换 π,计算图 G 的随机同构拷贝图 $G' = G\pi$,设图 G' 的邻接矩阵为 M',用 BC 对 M' 逐比特进行掩盖,得一新矩阵 M'',发送 M'' 给 V.

(2) V 随机选择 $b = 0$ 或 1,发送 b 给 P.

(3) P 检查,如果 $b = 0$,P 将 π、M' 发送给 V;如果 $b = 1$,P 将 M'' 中的与 M 的哈密顿回路对应的哈密顿回路揭开后发送给 V.

(4) V 进行验证,如果 $b = 0$,V 通过图 G、π 来验证 M' 是否正确,也就是检查 P 在(1)中是否按照协议要求来执行的;如果 $b = 1$,V 核实 M'' 的揭开部分是否构成一条哈密顿回路,以此来证明 P 是否知道 G 的一条哈密顿回路.对 M'' 的揭开部分应该是每行每列各揭开一个比特,并且均为1.

以上协议可重复执行 n 圈,以降低 P 欺骗的概率.

如果图的同构问题是困难的,以上协议所用的BC掩盖技术是安全的,那么上面的交互式证明协议就是一个零知识证明协议.

验证者 V 要么得到图 G 的一个同构拷贝,这一点他自己也可以做,要么只得到图的哈密顿回路,至于这个图是什么他也不知道.

另有一点需要注意,即在以上协议中,V 发送 b 给 P,向 P 索取一个概率分布上的数据,这一点 P 是无法控制的,权力在 V 手中.

由此可以相信有以下定理:

定理1 如果有安全的BC信道,那么就存在零知识证明协议.

6. 遗忘传递(oblivious transfer)

遗忘传递技术是指 A 向 B 传递两个状态 a、b,B 只能得到其中的一个,得到哪

个 A 无法确定，B 得到任何一个的概率为1/2.

传递两个素数 p、q 的遗忘传递协议是，A 送 p、q 给 B，要么 B 收到 p、q，要么什么也得不到.

(1) A 把两个素数的乘积 $n = pq$ 发送给 B.

(2) B 随机选择 $x < n$，$(x, n) = 1$，B 计算 $a = x^2 (\bmod n)$，发送 a 给 A.

(3) A 因为知道 p、q，A 计算 a 的4个平方根 x、$n - x$、y、$n - y$，发送其中一个给 B.

(4) 如果 B 收到 y 或 $n - y$，他计算 $x + y$ 和 n 的最大公因子 d，该公因子 d 就是 p 或 q，然后进而得到另一个 n/d；

如果 B 得到的是 x 或 $n - x$，B 什么也计算不出来.

6.7 量子密码

量子密码是利用光子的偏振特性实现的一种基于物理学的密码体系. 下面是一个产生秘密密钥的协议例子.

设 A 与 B 采用对称密钥密码体制进行保密通信，他们必须协商一个共同的密钥.

(1) A 把一串光子脉冲发送给 B，其中每个脉冲都是随机地在4个方向的一个方向上偏振，4个方向为：垂直 |，水平 __，左对角 \，右对角 /. 例如 A 发送的是：‖/—\—|—/.

(2) B 有一个偏振检测器，B 随机地摆设其检测器，记录检测结果，例如 ×++×××+×++，× 表示斜放，+ 表示正放. 可能的检测结果为：‖/—\—|—/.

(3) B 在公开信道上告诉 A 他使用的检测器设置.

(4) A 在公开信道上告诉 B 哪些是正确设置，如例子中的2,6,7,9是对的.

(5) A 与 B 仅保留这些正确设置的位子，例如 */* * */—*—*.

(6) 如果 A、B 事先规定 —、\ 为1，/、| 为0，那么他们将获得共同的密钥0011.

6.8 安全选举

通过综合运用以上各协议，可以设计较为复杂的选举协议，理想的选举协议应该满足以下5个条件：

(1) 只有经过许可的投票者才能投票，例如登记.

(2) 每个人只能投票一次.

(3) 任何人都不能确定别人投了谁的票.

(4) 任何人不能修改其他人的选票而不被发现.

(5) 所有投票人都能确认他们的票被统计在了最后的结果上了.

6.9 密钥管理与分配

在一个密码系统中,密钥的产生、分配、管理需要一整套措施来实施,密钥管理中心(KDC)往往起着关键作用.

在对称密钥密码系统中,KDC掌握与所有用户进行保密通信的密钥,用户之间的保密通信的密钥(工作密钥)往往由KDC分配,对用户身份的鉴别也由KDC出示的证书来鉴别.例如Kerberos系统就是一个典型的对称密钥密码体制的KDC管理系统.

在公开密钥密码系统中,往往需要建立一个认证中心(CA)来保证用户的公开密钥的合法性和真实性,而用户的公开密钥和秘密密钥往往由用户产生.CA对用户的身份确定后,为用户颁发证书,并对用户的公开密钥签名,所有用户的公开密钥存放在CA的公开数据库中,CA保证数据库的安全,防止非法修改,保证用户的真实性,并为可能的纠纷提供仲裁服务.

总之,信息安全的内容极其丰富,大量技术细节不能在此详述,详细内容可参考有关教材和文献,利用密码技术可以实现电子信息系统中的各种安全策略,保证数据的安全.

参 考 文 献

1 [美]施奈尔著.应用密码学.1996.

2 [美]*D. R.*斯廷森著,张文政译.密码学.国防科学技术保密通信重点试验室,1997.

3 卢开澄编著.计算机密码学.北京:清华大学出版社,1998.

4 王育民,何大可编著.保密学.西安:西安电子科技大学出版社,1990.

5 杨义先,林须端编著.编码密码学.北京:人民邮电出版社,1993.

·计算机数学卷·

第 24 篇

多值逻辑

编　者　刘任任
审校者　罗铸楷

目　录

引　言

多值逻辑是由二值逻辑扩展而来的.经典的二值逻辑只有两个状态,即“真”和“假”,任何命题“非真即假”,二者必居其一,即排中律成立.然而,客观世界的事物是十分复杂的,有些事物在某些情况下不是二值逻辑所能完全描述的.于是,便产生了多值逻辑.

多值逻辑的思想可以溯源到19世纪的苏格兰学者麦克柯尔(MacColl).但正式作为一种逻辑系统,则是20世纪由波兰逻辑学家卢卡西维茨(Lukasiewicz)和美国数学家波斯特(Post)分别于1920年和1921年各自独立提出来的.

多值逻辑的研究内容大体可分为三个方面,即多值逻辑的数学理论、多值电路与多值数字系统、多值逻辑的应用.

目前的计算机科学中主要使用的是二值逻辑.但三值、四值以及更高值的逻辑已在计算机科学中得到应用,并且正越来越多地渗透到计算机科学技术的许多分支中,显示着多值逻辑的强大生命力.例如,陈廷槐教授提出的四值逻辑与星算法,在组合电路与时序电路的测试生成、冗余线故障检测以及研究各种变化和转移现象(转移逻辑)等方面得到了很好的应用.

本篇主要介绍多值逻辑代数系统和多值逻辑函数的结构理论.关于多值逻辑的演算部分请参见近代数学卷第1篇的非经典逻辑部分.

1　多值逻辑代数系统

设S是一个n元集合,$n \geq 2$.定义在S上且函数值仍属于S的函数,称为**n值逻辑函数**.一组运算称为是**功能完备的**,如果定义在S上的任何一个n值逻辑函数都可以用这些运算表示出来.由这些运算组成的集合称为**功能完备集**.

以下设真值集合为$S = \{0,1,2,\cdots,n-1\}$,在S上定义相应的运算,便得到各种n值逻辑代数系统,它们可应用于电路分析和开关理论.

1.1　波斯特的n值系统

在S上定义一元运算为

$$\sim x = (x+1)\ (\mathrm{mod}\ n),$$

其真值表如表1-1.

表 1-1

x	0 1 … $n-2$ $n-1$
$\sim x$	1 2 … $n-1$ 0

二元运算为

$$x \vee y = \max(x, y).$$

对任何基数 n，以上两个运算是功能完备的，但函数的标准展开式很复杂.

1.2 阿伦(Allen)和吉汪(Givone)系统

在 S 上定义二元运算为

$$x + y = \max(x, y),$$
$$x \cdot y = \min(x, y),$$

一元运算定义为文字运算：

$$^{a}x^{b} = \begin{cases} n-1, & a \leqslant x \leqslant b, \\ 0, & \text{其它}, \end{cases}$$

式中 a、$b \in S$.

一个 n 值 m 变量逻辑函数 $F(x_1, x_2, \cdots, x_m)$ 的真值表共有 n^m 行. $F(x_1, x_2, \cdots, x_m)$ 可展开为下面的"积之和"形式：

$$F(x_1, x_2, \cdots, x_m) = \sum_{t=0}^{n^m-1} F(a_t)\, {}^{i}x_1^{i}\, {}^{j}x_2^{j} \cdots {}^{k}x_m^{k}$$

其中 $i, j, \cdots, k$ 为 $F(x_1, x_2, \cdots, x_m)$ 的真值表中第 t 行各变量的相应取值，即 $a_t = (i, j, \cdots, k)$，$\sum$ 为逻辑和"+".

以上三个运算加上 S 中的常数构成了一个功能完备集.

1.3 乌兰尼西(Vranesic)，李(Lee)与史密斯(Smith)系统

定义二元运算为

$$x + y = \max(x, y),$$
$$x \cdot y = \min(x, y),$$

一元运算有 $2n$ 个，其中 n 个一元反相运算为

$$x^{k} = \begin{cases} k, & x = 0, \\ 0, & x \neq 0, \end{cases}$$

$k = 0, 1, \cdots, n-1$；另外 n 个反循环运算为

$$\overset{\overset{k}{\leftarrow}}{x} = (x - k) \pmod{n},$$

$k = 0, 1, \cdots, n-1$.

对于定义在 S 上的任意一个 n 值 m 变量函数，其"和之积之和"展开式为

$$F(x_1, x_2, \cdots, x_m) = \sum_{j=1}^{n-1} \{ \prod_{F(u_1,\cdots,u_m)=j} (\sum_{i=1}^{m} \overset{u_i}{\underset{x_i}{\leftarrow}}) \}^j,$$

其中$\sum$为逻辑和“+”，$\prod$为逻辑积“·”. 因此，以上 $2n+2$ 个运算构成功能完备集.

1.4　模代数系统

在 S 上定义两个二元运算：

(1) 模 n 算术和：$(x+y) \pmod n$，

(2) 模 n 算术积：$(x \cdot y) \pmod n$.

对任何质数 n，以上两个运算加上逻辑常数后是功能完备的. 这是因为定义在 S 上的任意一个 n 值 m 变量函数 $F(x_1, x_2, \cdots, x_m)$ 可唯一地表示为标准式：

$$F(x_1, x_2, \cdots, x_m) = \sum_{i=1}^{n^m-1} (b_i \cdot X_i) \pmod n,$$

式中 X_i 为 $x_1, x_2, \cdots, x_m$ 的乘幂的一切可能的不同组合：

$$\prod_{i=1}^{m} x_i^{\alpha_i} = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_m^{\alpha_m}, \quad \alpha_i = 0, \cdots, n-1,$$

而 $b_i \in S$ 为各个 X_i 相应的系数.

例如，三值二变量函数的展开式为

$$F(x_1, x_2) = \{ b_{00} + b_{10} \cdot x_1 + b_{20} x_1^2 + b_{01} \cdot x_2 + b_{11} \cdot x_1 \cdot x_2 + b_{21} \cdot x_1^2 \cdot x_2 + b_{02} \cdot x_2^2 + b_{12} \cdot x_1 \cdot x_2^2 + b_{22} \cdot x_1^2 \cdot x_2^2 \} \pmod 3.$$

今设 $F(x_1, x_2)$ 的真值表如表 1-2.

表 1-2　$F(x_1, x_2)$ 的真值表

x_1	x_2	$F(x_1, x_2)$
0	0	0
0	1	0
0	2	0
1	0	0
1	1	0
1	2	2
2	0	2
2	1	1
2	2	2

根据真值表 1-2 可确定展开式中的系数如下：

$$b_{00}=0,\quad b_{10}=2,\quad b_{20}=1,$$
$$b_{01}=0,\quad b_{11}=2,\quad b_{21}=0,$$
$$b_{02}=0,\quad b_{12}=0,\quad b_{22}=1.$$

因而所求的三值二变量函数展开式为

$$F(x_1,x_2)=\{2\cdot x_1+x_1^2+2\cdot x_1\cdot x_2+x_1^2\cdot x_2^2\}\ (\mathrm{mod}\ 3).$$

1.5　韦布运算系统

在 S 上定义二元运算为

$$x_1\mid x_2=\begin{cases}(1+x_1)\ (\mathrm{mod}\ n), & x_1=x_2,\\ 0, & x_1\neq x_2.\end{cases}$$

此运算称为韦布(Webb)运算. 当 $n=2$ 时,韦布运算就是"或非"运算,其二值韦布运算的真值表如表 1-3.

表 1-3　韦布运算真值表

x_1	x_2	$x_1\mid x_2$
0	0	1
0	1	0
1	0	0
1	1	0

它是功能完备的.

2　阈值逻辑

人们总是希望机器具备功能更强,但更可靠的性能,以便能以更少的空间和消耗来快速完成复杂的工作. 这一方面促进了优化二值设计的工作,另一方面也迫使人们研究信息密度更大的"高阶逻辑". 多值逻辑是一种理论上已证明可行的选择方案.

阈值逻辑是一种其每个阈元件中包含更多信息的逻辑系统,它作为神经功能的模拟被提出来,其名称是根据"阈元件的输出取决于输入变量的权与内部阈值相比较"而得来的.

1963 年,汉森(Hanson)提出了一种阈值操作并建立了一个完备的三值代数系统;之后,梅里尔(Merrill)提出了"权"的概念,建立了现在的阈值逻辑.

阈值逻辑的研究主要是沿着两个方向进行:一个方向是着重于阈值理论,寻找使一个三值函数成为阈值函数的充分条件,以及实现参数的确定;另一个方面则着重于三值阈值逻辑的应用.

2.1 二值阈值逻辑

m 变量二值阈值函数定义如下：

$$F(x_1,x_2,\cdots,x_m)=\begin{cases}1, & \sum_{i=1}^{m} w_i x_i \geq t,\\ 0, & 其它,\end{cases}$$

其中 $x_i \in \{0,1\}$ 表示输入变量，w_i 表示当 $x_i=1$ 时对 x_i 所加的权，t 为阈值. 理论上 w_i 及 t 可为任意实数，但实际应用中通常限定它们为正整数.

当 $w_1=\cdots=w_m=t=1$ 时，有"或门"：

$$F(x_1,x_2,\cdots,x_m)=x_1 \vee \cdots \vee x_m;$$

当 $w_1=\cdots=w_m=1, t=m$ 时，有"与门"：

$$F(x_1,x_2,\cdots,x_m)=x_1 \wedge \cdots \wedge x_m.$$

由此可见，阈门是包含"或门"和"与门"的一种广义形式的门.

2.2 三值阈值函数

定义 1 设真值集为 $V=\{-1,0,+1\}$，n 维向量 $X=(x_1,x_2,\cdots,x_n)\in V^n$，$x_i \in V$. **三值阈值函数**定义为 $F: V^n \to V$.

定义 2 对任意 $x \in V$，定义 x 的补为 $\bar{x}=-x$. 例如 $\bar{1}=-1$. 常定义 $V=\{\bar{1},0,1\}$.

定义 3 称函数 $\delta F=\overline{F}(\overline{X})$ 为 F 的**对偶函数**，其中 $\overline{X}=(\overline{x_1},\overline{x_2},\cdots,\overline{x_n})$. 如果 $F(X)=\overline{F}(\overline{X})$，即 $F(X)=\delta F$，则 F 称为**自对偶函数**.

定义 4 令 $V_{\bar{1}}=\{X \mid F(X)=\bar{1}\}$，$V_1=\{X \mid F(X)=1\}$，并设 $|V_{\bar{1}}|=m$，$|V_1|=p$.

定义 5 $\sigma_L F(X)$ 称为 $F(X)$ 的**低半对偶函数**，如果 $m \geq p$，且

$$\sigma_L F(X)=\begin{cases}-1, & F(X)=F(\overline{X})=0, m<\dfrac{3^n}{2},\\ 0, & F(X)=F(\overline{X})=-1, m>\dfrac{3^n}{2},\\ F(X), & 否则.\end{cases}$$

定义 6 $\sigma_U F(X)$ 称为 $F(X)$ 的**高半对偶函数**，如果 $p \geq m$，且

$$\sigma_U F(X)=\begin{cases}1, & F(X)=F(\overline{X})=0, p<\dfrac{3^n}{2},\\ 0, & F(X)=F(\overline{X})=1, p>\dfrac{3^n}{2},\\ F(X), & 否则.\end{cases}$$

注意，有 $\sigma_L\sigma_L F(X)=F(X)$ 和 $\sigma_U\sigma_U F(X)=F(X)$.

定义 7　称 $F(X)$ 对 x_i 是单调递增(递减)的,如果对变量 $x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n$ 的任意取值,有

$$F_{i1} \geqslant F_{i0} \geqslant F_{i\bar{1}} \quad (F_{i1} \leqslant F_{i0} \leqslant F_{i\bar{1}}),$$

其中 $F_{ik} = F(x_1, x_2, \cdots, x_{i-1}, k, x_{i+1}, \cdots, x_n), k \in V$.

定义 8　如果对每个变量 $x_i (i = 1,2,\cdots,n)$, $F(X)$ 是单调递增(递减)的,则称 $F(X)$ 是单调递增(递减)函数.

定义 9　如果对每个变量 $x_i (i = 1,2,\cdots,n)$, $F(X)$ 是单调递增或递减的,则称 $F(X)$ 是单调函数.

定义 10　称 $F(X)$ 为对变量 x_i、x_j 是对称的,如果交换 x_i、x_j 的位置函数值保持不变,即,

$$F(x_1, \cdots, x_i, \cdots, x_j, \cdots, x_n) = F(x_1, \cdots, x_j, \cdots, x_i, \cdots, x_n).$$

$F(X)$ 称为对称函数,如果对任意变量 x_i、$x_j (1 \leqslant i, j \leqslant n)$, 函数 $F(X)$ 是对称的.

定义 11　$F(X)$ 称为**阈值函数**,如果存在一权向量 $w = (w_1, w_2, \cdots, w_n)$ 和非空阈值集 $B = \{H, L\} (H \geqslant L)$,使得

$$w \cdot X \geqslant H \Leftrightarrow F(X) = 1,$$

$$H > w \cdot X > L \Leftrightarrow F(X) = 0, \qquad w \cdot X = \sum_{i=1}^{n} w_i x_i.$$

$$L \geqslant w \cdot X \Leftrightarrow F(X) = \bar{1},$$

如果 $F(X)$ 是一阈值函数,则 (w, H, L) 称为其**实现集**,记为 $F(X):(w, H, L)$.

定义 12　设 $F(X)$ 是一个 n 元函数,它的**特征参数向量**(c 参数)是如下 $n+2$ 维向量:

$$c = (c_1, c_2, \cdots, c_n, p, m),$$

其中, $c_i = \sum\limits_{X \in V^n} x_i F(X)$.

如果 $F(X)$ 是一阈值函数,则在 c 参数和函数之间存在一一对应关系,记作 $F(X) \to (c, p, m)$.

定义 13　设 X_1、X_2、X_3、X_4 为输入向量,如果 $X_1 - X_2 = X_3 - X_4$,则说输入向量对偶 (X_1, X_2) 和 (X_3, X_4) 是平行的,记作 $(X_1, X_2) /\!/ (X_3, X_4)$. 称 $F(X)$ 是完全单调的,如果

$$[F(X_1) - F(X_2)] \cdot [F(X_3) - F(X_4)] \geqslant 0$$

对所有具有平行关系的输入向量对偶均成立.

定义 14　设 M 是一个 n 元多值逻辑函数集. 称 M 是完全单调的,如果

1° 每个 $F_i \in M$ 是完全单调的.

2° 对每对函数 F_i、$F_j \in M$ 以及具有平行关系的输入向量对偶 (X_1, X_2)、(X_3, X_4),满足

$$[F_i(X_1) - F_i(X_2)] \cdot [F_j(X_3) - F_j(X_4)] \geqslant 0.$$

定理 1　任一阈值函数是单调的.

定理 2　设 $F(X)$ 是一阈值函数, $F(X):(w, H, L)$, 于是,以下 $F(X)$ 的函数

亦是阈值函数(右边是相应实现集):

$$F(x_1,\cdots,\overline{x_i},\cdots,x_n):(w_1,\cdots,-w_i,\cdots,w_n,H,L),$$
$$F(\cdots,x_n,\cdots,x_i):(\cdots,w_n,\cdots,w_i,H,L),$$
$$\overline{F}(x_1,x_2,\cdots,x_n):(-w_1,-w_2,\cdots,-w_n,-L,-H),$$
$$\delta F(X):(w,-L,-H),$$
$$\sigma_L F(X):(w,H,-L),$$
$$\sigma_U F(X):(w,-H,L).$$

推论 1 如果 $F(X):(w,H,L)$,则 $F(X)$ 是一个自对偶函数.

定理 3 一个变量数目 $n\leqslant 3$ 的三值函数是阈值函数的充分必要条件是它是完全单调函数.

定理 4 设 $F(X):(w,H,L)\rightarrow(c_1,c_2,\cdots,c_n,p,m)$.以下 $F(X)$ 的函数亦是阈值函数(右边是相应特征参数):

$$F(x_1,\cdots,\overline{x_i},\cdots,x_n)\rightarrow(c_1,\cdots,-c_i,\cdots,c_n,p,m),$$
$$F(\cdots,x_j,\cdots,x_i,\cdots)\rightarrow(\cdots,c_j,\cdots,c_i,\cdots,c_n,p,m),$$
$$\overline{F}(X)\rightarrow(-c_1,\cdots,-c_n,m,p),$$
$$\delta F(X)\rightarrow(c_1,\cdots,c_n,m,p),$$
$$\sigma_L F(X)\rightarrow(c_1,\cdots,c_n,p,3^n-m),$$
$$\sigma_U F(X)\rightarrow(c_1,\cdots,c_n,3^n-p,m).$$

定理 5 设 $F(X):(w,H,L)\rightarrow(c_1,c_2,\cdots,c_n,p,m)$.于是

1° $c_i\neq 0\Rightarrow w_i$ 与 c_i 同号,

$c_i\neq c_j\Rightarrow c_i>c_j\Leftrightarrow w_i>w_j$.

2° $c_i=0\Rightarrow w_i=0$,

$c_i=c_j\Rightarrow w_i=w_j$.

3° $p>3^n/2\Leftrightarrow H<0$,

$m>3^n/2\Leftrightarrow L>0$.

4° $p>m\Leftrightarrow |H|\geqslant -L$,

$p<m\Leftrightarrow H\geqslant |L|$,

$p=m\Leftrightarrow H=-L$.

定义 15 称阈值函数 $F(X)$ 是可接受正则函数(简称为 AC 函数),记为 $F(X)ac$,如果 $F(X)$ 具有如下性质:

1° $w_1\geqslant w_2\geqslant w_3\cdots\geqslant w_n>0\ (\Leftrightarrow c_1\geqslant c_2\geqslant c_3\cdots\geqslant c_n>0)$.

2° $m>3^n/2\ (\Leftrightarrow H\geqslant L>0)$.

定理 6 设 $F(X)$ 是一个满足 $p+m=3^n$ 的阈值函数,即 $F(X)\in\{\overline{1},1\}$,则

$$F(X)ac\rightarrow(c_1,\cdots,c_n,p,3^n-p):(w_1,\cdots,w_n,L,L)$$
$$\Leftrightarrow \alpha F(X)ac\rightarrow\left(\frac{c_1}{2},\cdots,\frac{c_n}{2},0,3^n-p\right):(w_1,\cdots,w_n,\infty,L),$$

其中

$$\alpha: F(X) \to V, \alpha(\bar{1}) = 1,\quad \alpha(1) = 0,$$
$$\alpha^{-1}: \alpha F(X) \to V, \alpha^{-1}(\bar{1}) = \bar{1},\quad \alpha^{-1}(0) = 1.$$

定理 7　设 $F(X)$ 是一个阈值函数，且 $p = 0$，则

$$F(X)ac \to (c,0,m):(w,\infty,L) \Leftrightarrow \beta F(X)ac \to (c',1,m):(w,H,L),$$

其中
$$\beta F(X) = \begin{cases} F(X), & X \neq (1,1,\cdots,1), \\ 1, & X = X_0 = (1,1,\cdots,1), \end{cases}$$
$$c'_i = c_i + 1, i \leqslant n,\quad H = \sum_i (w_i - \frac{1}{2} w_n).$$

定理 8　设 $F(X)$ 是一个满足 $p + m = 3^n$ 的阈值函数，即 $F(X) \in \{\bar{1},1\}$，则

$$F(X)ac \to (c,3^n - m,m):(w,L,L) \Leftrightarrow \beta F(X)ac \to (c'',1,m):(w,H,L),$$

其中 $\beta F(X)$ 定义见定理 7，且

$$c''_i = \frac{1}{2}c_i + 1, H = \sum_i (w_i - \frac{1}{2} w_n),\quad i \leqslant n.$$

定理 9　设 $F(X)$ 是一个满足 $p = 1, c_{n-1} \neq c_n$ 的阈值函数，则

$$F(X)ac \to (c,1,m):(w,H,L) \Leftrightarrow \lambda F(X)ac \to (c''',2,m):(w,H',L)$$

其中
$$\lambda F(X) = \begin{cases} F(X), & X \neq (1,1,\cdots,1,0), \\ 1, & X = (1,1,\cdots,1,0), \end{cases}$$

且对所有 $i \neq n, c'''_i = c_i + 1, c'''_n = c_n, H' = H - w_n$.

定理 10　设 $F(X)$ 是一个满足 $c_{n-1} \neq c_n + 1, p = 2$ 的阈值函数，则

$$F(X)ac \to (c_1,\cdots,c_{n-1},c_n,2,m):(w,H,L)$$
$$\Leftrightarrow \psi F(X)ac \to (c'_1,\cdots,c'_{n-1},c'_n,3^n - m,m):(w,L,L),$$

其中

$$H = \sum_{i=1}^{n-1} (w_i - \frac{1}{2} w_n),$$
$$\psi: F(X) \to V, \psi(\bar{1}) = -1,\quad \psi(0) = \psi(1) = 1,$$
$$\psi^{-1}: \psi F(X) \to V,\quad \psi^{-1}(-1) = -1,$$
$$\psi^{-1}(1) = \begin{cases} 0, & X \neq (1,1,1,\cdots,1,1) \text{ 且 } X \neq (1,1,1,\cdots,1,0), \\ 1, & X = (1,1,1,\cdots,1,1) \text{ 或 } X = (1,1,1,\cdots,1,0), \end{cases}$$

且对所有 $i < n, c'_i = 2(c_i - 2), c'_n = 2(c_n - 1)$.

3　多值逻辑函数的结构理论

在多值逻辑理论中，函数系完备性的判定问题是一个基本而重要的问题．同时也是自动机理论、多值逻辑网络中必须解决的问题，此问题的彻底解决依赖于定出多值逻辑函数集中的所有极大封闭集（又称准完备集）．它包含着三个著名的问题，即分别定出完全多值逻辑函数集、部分多值逻辑函数集、一元多值逻辑函数集中的所有极大封闭集．

3.1 完全多值逻辑函数

设 $E_k=\{0,1,\cdots,k-1\}$, $k\geqslant 2$. 函数 $f(x_1,x_2,\cdots,x_n):E_k^n\to E_k$ 称为**完全 k 值逻辑函数**. 所有完全 k 值逻辑函数作成的集合记为 P_k.

定义 1 设 $\varnothing\subset A\subseteq P_k$, 函数 $f(t_1,t_2,\cdots,t_m)\in A$; 函数 $g_i(x_1,x_2,\cdots,x_n)$ $(i=1,2,\cdots,m)$ 或者属于 A 或者为自变数 x_i. 称函数 $f(g_1(x_1,x_2,\cdots,x_n),\cdots,g_m(x_1,x_2,\cdots,x_n))\in A$ 是由 A 中的函数复合出来的函数. 所有由 A 中的函数复合出来的函数作成之集合记为 A'.

由定义易知, $A\subseteq A'$ 并且若 $A\subseteq B$, 则 $A'\subseteq B'$.

定义 2 设 $\varnothing\subset A\subseteq P_k$. 如果 $A'=A$, 则称 A 是一个**封闭集**.

不难知道, 任意有限多个封闭集的交集仍是封闭集.

定义 3 设 $\varnothing\subset E\subseteq E_k$, $f(x_1,x_2,\cdots,x_m)\in P_k$. 若对任意 $\alpha_1,\alpha_2,\cdots,\alpha_m\in E$ 均有 $f(\alpha_1,\alpha_2,\cdots,\alpha_m)\in E$, 则称函数 $f(x_1,x_2,\cdots,x_m)$ 是**保 E 的**. 所有保 E 函数之集合记为 T_E, 常将 $T_{\{i\}}$ 简记为 T_i, $i\in E_k$.

T_E 是封闭集.

设 $E_k=A_1\cup A_2\cup\cdots\cup A_m$ 是 E_k 的一个直接划分, 即 $A_i\neq\varnothing$, $A_i\cap A_j=\varnothing$, $i\neq j$, $i,j=1,2,\cdots,m$, 将此直接划分记为 DP: $A_1+A_2+\cdots+A_m$.

定义 4 设 DP: $A_1+A_2+\cdots+A_m$ 是 E_k 的一个直接划分, 如果对任意 $(\alpha_1,\alpha_2,\cdots,\alpha_n)$、$(\beta_1,\beta_2,\cdots,\beta_n)$, $\alpha_i,\beta_i\in A_{h_i}$ $(1\leqslant h_i\leqslant m,\ i=1,2,\cdots,n)$, 有 A_h $(1\leqslant h\leqslant m)$ 使得 $f(\alpha_1,\alpha_2,\cdots,\alpha_n)$、$f(\beta_1,\beta_2,\cdots,\beta_n)\in A_h$, 则称函数 $f(x_1,x_2,\cdots,x_m)$ 是**保 DP 的**. 所有保 DP 函数之集合记为 T_{DP}.

T_{DP} 也是封闭集.

定义 5 设 $\varnothing\subset A\subseteq P_k$, 称

$$\mathrm{GEN}(A)=A\cup A'\cup(A')'\cup\cdots$$

为 A 的**封化集**.

$\mathrm{GEN}(A)$ 中的函数称为 A 的叠合函数, 它们都是由 A 中的函数经过有限次复合而产生出来的函数.

定义 6 设 $\varnothing\subset A\subseteq P_k$. 若 $\mathrm{GEN}(A)=P_k$, 则称 A 是完备集. 特别, 若 $\mathrm{GEN}(\{f(x_1,x_2,\cdots,x_n)\})=P_k$, 则称 $f(x_1,x_2,\cdots,x_n)$ 为**谢弗(Sheffer)函数**.

例 1 函数

$$f(x,y)=\max(x,y)+1\ (\mathrm{mod}\ k)$$

是谢弗函数.

定义 7 设 $M\subset P_k$ 是一个封闭集, 若 M 与 P_k 之间无另外的封闭集, 则称 M 为一个极大封闭集或准完备集.

定理 1 设 $M\subset P_k$, M 是极大封闭集, 当且仅当对任意 $f\notin M$, 有

$$\mathrm{GEN}(M\cup\{f\})=P_k.$$

定理 2 设 $\varnothing\subset E\subset E_k$, 则 T_E 是极大封闭集.

关于函数集 A 的完备性判定,有

定理 3 P_k 中的非空子集 A 是完备集,当且仅当 A 不含于任何极大封闭集.

此定理说明,只要定出 P_k 中的所有极大封闭集,则函数集完备性之判定问题也就完全解决了.

波斯特 1941 年定出了 $k = 2$ 时 P_2 中的所有极大封闭集共 5 个;

雅布朗斯基(Jablonskii)1958年定出了 $k = 3$ 时 P_3 中的所有极大封闭集共18个.

对于一般的 $k \geqslant 3$,雅布朗斯基和马丁朱克(Martynjuk)于 1958 年、1960 年分别定出了自对偶函数 S_σ、T 型函数集 $T_{E,0}$、单调函数集 M 中的所有极大封闭集,罗铸楷教授于 1963 年定出了线性函数集 L_G 中的所有极大封闭集,保划分函数集 T_{U} 中的大量极大封闭集(仅剩一类尚未定出),并于 1964 年证明了 P_k 中的任意极大封闭集必是以上五个类型的函数集之一.罗森贝格(Rosenberg)于 1965 年也得出了此结论,并定出了 T_{U} 中的所有极大封闭集.

设 $\Sigma = \{A \mid A \subset P_k$ 是极大封闭集$\}$,$\Sigma' \subseteq \Sigma$.令

$$S_{\Sigma'} = \{f \in P_k \mid f \in A' \in \Sigma'\} = \bigcup_{A' \in \Sigma'} A'.$$

定义 8 称集合 $\Sigma'(\subseteq \Sigma)$ 是 Σ 的覆盖,如果 $S_{\Sigma} = S_{\Sigma'}$.称覆盖 Σ' 是**最小覆盖**,如果 Σ' 的任何一个真子集都不是 Σ 的覆盖.

定理 4 $f \in P_k$ 是谢弗函数,当且仅当对任何 Σ 的覆盖 Σ',$f \in S_{\Sigma'}$.

显然,对于一个极大封闭集 A,若存在一个函数 $f \in A$,使得 f 不属于除 A 以外的任何一个极大封闭集,则 A 必是最小覆盖的组成部分.

3.1.1 二值逻辑函数集

众所周知,当 $k = 2$ 时,$E_2 = \{0,1\}$ 是一个布尔代数.在布尔代数中,$xy = \min(x,y)$,$x + y = \max(x,y)$,$\bar{x} = 1 - x$.

P_2 中的函数集$\{0,1,xy,x \oplus y\}$ 是完备的,其中 $x \oplus y = x\bar{y} + \bar{x}y = x + y \bmod 2$.于是,任意一个二值逻辑函数可表示为 $x_1, x_2, \cdots, x_n$ 的多项式(模 2 相加).

定义 9 称 $f(x_1,x_2,\cdots,x_n) \in P_2$ 是线性函数,如果 $f(x_1,x_2,\cdots,x_n) = a_0 \oplus a_1x \oplus \cdots \oplus a_nx_n$,其中 $a_i \in E_2$,$i = 0,1,\cdots,n$.所有线性函数之集合记为 L.

定义 10 称 $f(x_1,x_2,\cdots,x_n) \in P_2$ 是单调函数,如果对任意 $\tilde{\alpha} = (\alpha_1,\alpha_2,\cdots,\alpha_n) \leq \tilde{\beta} = (\beta_1,\beta_2,\cdots,\beta_n)$(即 $\alpha_i \leqslant \beta_i$,$i = 1,2,\cdots,n$)均有 $f(\tilde{\alpha}) \leqslant f(\tilde{\beta})$,$\alpha_i$、$\beta_i \in E_2$,$i = 1,2,\cdots,n$.所有单调函数之集合记为 M.

定义 11 称 $f(x_1,x_2,\cdots,x_n) \in P_2$ 是自对偶函数,如果 $f(x_1,x_2,\cdots,x_n) = \bar{f}(\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_n)$.所有自对偶函之集合记为 S.

定义 12 称 $f(x_1,x_2,\cdots,x_n)$ 是保 i 函数,如果 $f(i,i,\cdots,i) = i$.所有保 i 函数之集合记为 T_i,$i \in E_2$.

以上定义中的五个函数集均是极大封闭集.

定理 5(波斯特完备性定理) P_2 中的非空子集 A 是完备集,当且仅当 A 不含

于上述五个封闭集的任何一个.

定义 5 若 $A \subseteq P_2$ 完备且 A 的任何真子集均不完备,则称 A 是 P_2 的一个底.

例 2 $x \oplus y, xy, 1; x + y, \bar{x}; xy, \bar{x}$ 以及 $0, 1, x \oplus y \oplus z, xy$ 都是 P_2 的底.

定理 6 P_2 中的任意完备集中必包含一个底,其中至多有 4 个函数.

定理 7 $\{T_0, T_1, S\}$ 是 $\{L, M, T_0, T_1, S\}$ 的最小覆盖.

例 2 令 $f(x,y) = \begin{cases} 0, & (x,y) = (0,1),(1,0),(1,1), \\ 1, & (x,y) = (0,0). \end{cases}$ 容易验证, $f \bar{\in} T_0 \cup T_1 \cup S$. 于是,由定理 3 及 3.1 节的定理 4 知, f 是一个谢弗函数.

3.1.2 $k(\geqslant 3)$ 值逻辑函数集

本节介绍 $P_k(k \geqslant 3)$ 中的极大封闭集.

定义 13 设 G 是一个 k 元加法群,称 $f(x_1, x_2, \cdots, x_n) \in P_k$ 是线性函数,如果对任意值组 $\tilde{\boldsymbol{\alpha}} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$, $\tilde{\boldsymbol{\beta}} = (\beta_1, \beta_2, \cdots, \beta_n)(\alpha_i, \beta_i \in G, i = 1, 2, \cdots, n)$, 均有

$$f(\tilde{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\beta}}) + f(\tilde{\mathbf{0}}) = f(\tilde{\boldsymbol{\alpha}}) + f(\tilde{\boldsymbol{\beta}}),$$

其中 $\tilde{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\beta}} = (\alpha_1 + \beta_1, \cdots, \alpha_n + \beta_n)$, $\tilde{\mathbf{0}} = (0, \cdots, 0)$.

所有线性函数之集合记为 L_G.

例如,常数 $a \in G$ 是线性函数, $f(x,y) = x + y$ 也是线性函数. L_G 是封闭集.

定理 8 设 G 是一个 k 元加法群. L_G 是极大封闭集,当且仅当 G 中的每一个元素(除 0 外)的周期均为同一个质数 p.

定义 14 设 σ 是集合 E_k 上的一个置换,称函数 $g(x_1, x_2, \cdots, x_n)$ 是函数 $f(x_1, x_2, \cdots, x_n)$ 关于 σ 的对偶函数,如果 $g(x_1, x_2, \cdots, x_n) = \sigma^{-1} f(\sigma(x_1), \sigma(x_2), \cdots, \sigma(x_n))$.

定理 9 设 G 是一个 k 元初等加法群, $k = p^n$, p 为质数; H 是所有线性置换作成的群. 则 P_k 中的所有极大线性封闭集为 $L_G, L_G^{\sigma_1}, \cdots, L_G^{\sigma_{m-1}}$, 其中 $\sigma_i \in H_i, i = 1, \cdots, m-1, m = p^n!/p^n(p^n - p^0)\cdots(p^n - p^{n-1})$, $H, H_1, \cdots, H_{m-1}$ 是 H 在 S_k 中的所有右陪集, S_k 是 G 上的 k 次对称群.

设二元关系"$\prec$"是定义在集合 E_k 上的偏序关系(又称部分序关系).

值组 $\tilde{\boldsymbol{\alpha}} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$, $\tilde{\boldsymbol{\beta}} = (\beta_1, \beta_2, \cdots, \beta_n)$ 说是 $\tilde{\boldsymbol{\alpha}} \prec \tilde{\boldsymbol{\beta}}$, 如果 $\alpha_i \prec \beta_i$, $\alpha_i, \beta_i \in E_k$, $i = 1, \cdots, n$.

定义 15 称 $f(x_1, x_2, \cdots, x_n)$ 是单调函数,如果对任意 $\tilde{\boldsymbol{\alpha}} \prec \tilde{\boldsymbol{\beta}}$, 均有 $f(\tilde{\boldsymbol{\alpha}}) \prec f(\tilde{\boldsymbol{\beta}})$. 所有单调函数之集合记为 M.

定理 10 单调函数集 M 是 P_k 中的极大封闭集,当且仅当偏序集 E_k 中恰有一个极大元和一个极小元.

令 $P(n)$ 表示 n 元偏序集的个数,于是恰有一个极大元和一个极小元的 k 元偏

序集之个数为 $k(k-1)P(k-2)$，从而，极大单调函数集之个数为$\binom{k}{2}P(k-2)$.

定义 16 设 σ 是E_k上的一个非恒等置换，称函数$f(x_1,x_2,\cdots,x_n)$是关于σ的自对偶函数，如果

$$f(x_1,x_2,\cdots,x_n)=\sigma^{-1}f(\sigma(x_1),\cdots,\sigma(x_n)),$$

即 $\sigma f(x_1,x_2,\cdots,x_n)=f(\sigma(x_1),\cdots,\sigma(x_n))$. 所有关于 σ 自对偶的函数之集合记为 S_σ.

例 3 令 $f(x,y)=x^2+y^2+(K-2)xy+x(\mathrm{mod}K)$、$\sigma(x)=x+1(\mathrm{mod}\ K)$，于是

$$\begin{aligned}f(\sigma(x),\sigma(y))&=f(x+1,y+1)\\&=(x+1)^2+(y+1)^2+(K-2)(x+1)(y+1)+x+1\\&=x^2+y^2+(K-2)xy+x+1\\&=f(x,y)+1=\sigma(f(x,y))\ (\mathrm{mod}\ K).\end{aligned}$$

因此，$f(x,y)$ 是关于 σ 自对偶的函数.

定理 11 S_σ 是P_k中的极大封闭集，当且仅当 σ 可分解为不相杂同一质数长度 p 的轮换之乘积.

定理 12 设 S_σ 和 S_τ 是极大封闭集，则 $S_\sigma=S_\tau$ 当且仅当有 h 使 $\tau=\sigma^h$.

设 $k=mp$，p 是质数. 称 $\sigma\in S_k$ 是一个(m,p)置换，如果 σ 能分解为 m 个长度为 p 的不相杂轮换之乘积.

由定义知，S_k 中所有(m,p)置换共有 $k!/(m!p^m)$个，从而由定理12，对给定的 m、p，共有 $k!/(m!p^m(p-1))$ 个不同的极大自对偶函数集. 于是有

定理 13 P_k 中的极大自对偶函数集之个数为

$$\sum_{k=mp}\frac{k!}{m!(p-1)p^m},$$

其中 p 为质数.

设划分 $D:E_k=A_1\cup\cdots\cup A_m$，称 E_k 中 s 个元素是保划分D的，如果 s 个元素同属于某个 A_i，$1\leqslant i\leqslant m$. 称 s 个值组 $\tilde{\alpha}^1=(\alpha_1^1,\alpha_2^1,\cdots,\alpha_n^1),\cdots,\tilde{\alpha}^s=(\alpha_1^s,\alpha_2^s,\cdots,\alpha_n^s)$ 是保划分 D 的，如果 $\alpha_i^1,\alpha_i^2,\cdots,\alpha_i^s$ 保划分 D，$i=1,2,\cdots,n$.

称 r 是划分 D 的阶数，如果 E_k 中任意 $r-1$ 个元素保划分 D，而存在 r 个不同的元素不保划分 D. 阶数为 r 的划分记为 D^r.

定义 17 称函数$f(x_1,x_2,\cdots,x_n)$是保划分 D^r 的，如果对任意 r 个保划分D^r的值组

$$\tilde{\alpha}^i=(\alpha_1^i,\alpha_1^i,\cdots,\alpha_n^i),\quad i=1,2,\cdots,r,$$

均有$f(\tilde{\alpha}^1),f(\tilde{\alpha}^2),\cdots,f(\tilde{\alpha}^r)$ 保划分 D^r. 所有保划分 D^r 的函数之集合记为 T_{D^r}.

定义 18 称 D^r 是中心划分，如果有 $a\in E_k$ 使得所有含a的r个元素均保划分 D^r.

由定义知，D^r 是中心划分，当且仅当 E_k 中所有不保划分 D^r 的 r 个元素之并不等于 E_k.

设 D 是 E_k 的一个直接划分，即 $D:E_k = A_1 + \cdots + A_r, A_i \cap A_j = \varnothing, i \neq j, i, j = 1,\cdots,r$. 称集合 $\{a_1, a_2, \cdots, a_r\}$ 为划分 D 所生成的子集，其中 $a_i \in A_i, i = 1,2,\cdots,r$.

定义 19　称划分 D^r 是正则的，如果存在 $m(\geqslant 1)$ 个直接划分

$$D_i: E_k = A_1^{(i)} + \cdots + A_r^{(i)}, \quad i = 1,\cdots,m,$$

使得不保划分 D^r 的所有 r 元子集恰为由 $D_1, D_2, \cdots, D_m$ 所生成的全部子集，且当 $m \geqslant 2$ 时，

$$A_{i1}^{(1)} \cap A_{i2}^{(2)} \cap \cdots \cap A_{im}^{(m)} \neq \varnothing, \quad 1 \leqslant i_1, i_2, \cdots, i_m \leqslant r.$$

定理 14　T_{D^r} 是极大封闭集（$r \leqslant 3$），当且仅当 D^r 是中心划分或正则划分.

若将 P_k 中所有保中心划分函数集（包括保 E 函数集）和保正则划分函数集的个数分别记为 $\lambda(D_C^r)$ 和 $\lambda(D_R^r)$，则

$$\lambda(D_C^r) = -k + \sum_{h=1}^{k}\sum_{i=1}^{k}(-1)^{i-1}\binom{k}{2}2^{\binom{k-i}{h}},$$

$$\lambda(D_R^r) = \sum_{\substack{h^m \geqslant k \\ h \geqslant 3, m \geqslant 1}} \frac{(-1)^h}{m!(h!)^m}\sum_{l=1}^{h^m}(-1)^l\binom{h^m}{l}l^k.$$

定理 15　当 $k \geqslant 3$ 时，P_k 中的全部极大封闭集是：线性函数集 L_G（其中 G 是一个初等加法群），单调函数集 M（其中 E_k 是一个恰有一个极大元和一个极小元的偏序集），自对偶函数集 S_σ（其中 σ 能分解为同一质数长度的不相杂轮换之乘积），保 E 函数集 T_E，保直接划分函数集 T_{DP}，保中心划分函数集 $T_{CD^r}(r \geqslant 2)$，保正则函数集划分 $T_{RD^r}(r \geqslant 3)$.

定理 16（完备性定理）　设 $A \subseteq P_k, A \neq \varnothing, k \geqslant 3$. A 是完备集，当且仅当 A 不包含在极大封闭集 L_G、M、S_σ、T_E、T_{DP}、$T_{CD^r}(r \geqslant 2)$、$T_{RD^r}(r \geqslant 3)$ 的任何一个之中，其中 G 是一个初等加法群，E_k 是一个恰有一个极大元和一个极小元的偏序集，置换 σ 能分解为同一质数长度的不相杂轮换之乘积.

令 $\Sigma' = \{S_\sigma, T_E, T_{DP}\}$，$S_{\Sigma'} = S_\sigma \cup T_E \cup T_{DP}$，则有

定理 17　(1) $f \in P_k$ 是谢弗函数，当且仅当 $f \bar{\in} S_{\Sigma'}$，即覆盖了所有的极大封闭集.

(2) 极大封闭集 M 包含一个函数不在其它的极大封闭集中，当且仅当 $M \subseteq S_{\Sigma'}$，即 $S_{\Sigma'}$ 是一个最小覆盖.

(3) 极大封闭集 M 中有 M 的谢弗函数，当且仅当 $M \bar{\subseteq} S_{\Sigma'}$.

3.2　部分多值逻辑函数

设 $E_k = \{0,1,\cdots,k-1\}, k \geqslant 2$. 函数 $f(x_1, x_2, \cdots, x_n): E_k^n \to E_k \cup \{*\}$ 称为**非完全 k 值逻辑函数**，其中 $f(\alpha_1, \alpha_2, \cdots, \alpha_n) = *$ 表示函数 f 在 $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ 处无定

义，$\alpha_i \in E_k$，$i = 1,2,\cdots,n$. E_k 上的完全和非完全 k 值逻辑函数统称为**部分 k 值逻辑函数**，所有部分 k 值逻辑函数作成之集合记为 P_k^*. 显然 $P_k \subset P_k^*$.

王湘浩教授用群论的方法对部分 k 值逻辑函数的完备性问题进行了研究，提出了一个完备性的充要条件，并依此条件定出了 P_2^* 和 P_3^* 中的所有极大封闭集. 弗雷瓦德(Freivald)证明了 P_2^* 中任意极大封闭集必是保某一个 k^2 项关系的函数集，也定出了 P_2^* 中的所有极大封闭集；前苏联学者波莫夫(Pomob)定出了 P_k^* 中一类特殊的极大封闭集，并依此定出了 P_3^* 中的所有极大封闭集. 对于一般的 $k \geq 4$，罗铸楷教授根据王湘浩教授提出的"保关系"的系统思想，定出了 P_k^* 中的所有极大封闭集，从而彻底解决了 P_k^* 中函数系完备性的判定问题.

3.2.1 基本概念

处处无定义的函数称为**空函数**，记为 $*$. 与完全 k 值逻辑函数一样，P_k^* 中也有复合(叠合)函数、(极大)封闭集、完备集、(最小)覆盖等概念.

定义 20 设 $f(t_1,t_2,\cdots,t_m)$、$g_i(x_1,x_2,\cdots,x_n) \in P_k^*$，$i = 1,2,\cdots,m$，复合函数 $h(x_1,x_2,\cdots,x_n) = f(g_1(x_1,x_2,\cdots,x_n),\cdots,g_m(x_1,x_2,\cdots,x_n))$ 定义如下：

对任意值组 $\tilde{\alpha} = (\alpha_1,\alpha_2,\cdots,\alpha_n)$，

$$h(\tilde{\alpha}) = \begin{cases} f(g_1(\tilde{\alpha}),\cdots,g_m(\tilde{\alpha}))， & 若\ g_1(\tilde{\alpha}),\cdots,g_m(\tilde{\alpha})\ 均有定义， \\ *， & 否则. \end{cases}$$

设 $A \subseteq P_k^*$，$f(t_1,t_2,\cdots,t_m) \in A$，$g_i(x_1,x_2,\cdots,x_n) \in A \cup \{x_1,x_2,\cdots,x_n\}$，$i = 1,2,\cdots,m$. 函数 $f(g_1(x_1,x_2,\cdots,x_n),\cdots,g_m(x_1,x_2,\cdots,x_n))$ 称为由 A 中的函数复合出来的函数，简称由 A 复合出来的函数.

定义 21 所有由 A 复合出来的函数作成之集合记为 A'. 如果 $A = A'$，则称 A 是一个封闭集.

定义 22 设 $A \subseteq P_k^*$，令 $GEN(A) = A \cup A' \cup (A')' \cup \cdots$. $GEN(A)$ 中的函数称为 A 的**叠合函数**，它是由 A 中的函数经过有限次复合而成.

定义 23 设 $A \subseteq P_k^*$，如果 $GEN(A) = P_k^*$，则称 A 是一个完备集.

定义 24 若 $A = \{f\}$ 是完备集，则 f 称为谢弗函数，即由 f 可叠合出 P_k^* 中的全部函数.

例 4 函数

$$f(x,y,z) = \begin{cases} \max(x,y) + 1 \bmod k, & y = z, \\ *, & y \neq z \end{cases}$$

是谢弗函数.

定义 25 设 M 是一个封闭集，$M \subset P_k^*$. 若 M 与 P_k^* 之间无另外的封闭集，则称 M 为一个极大封闭集或准完备集.

定理 18 设 $A \subseteq P_k^*$，且 A 能复(叠)合出 P_k 中的所有函数. 若 A 中有一个非空非完全的函数，则 A 完备.

例 5 若 $A \subseteq P_k^*$ 在 P_k 中完备，则 A 添上任意一个非空非完全函数后，便在 P_k^*

中完备.

定理 19 包含全部完全二元函数的极大封闭集只有一个,即 $P_k \cup \{*\}$,其中 $*$ 是空函数.

3.2.2 极大封闭集

令 $G_m = \{\langle a_1, a_2, \cdots, a_m\rangle \mid a_i \in E_k, i = 1,2,\cdots,m\} \subseteq E_k^m, m \geq 2$,称 G_m 为 E_k 上的一个 **m 项关系**.

定义 26 称函数 $f(x_1, x_2, \cdots, x_n)(\in P_k^*)$ 是**保 m 项关系 G_m 的**,如果对任意保 G_m 的 n 元值组 $\tilde{\alpha}^1, \tilde{\alpha}^2_2, \cdots, \tilde{\alpha}^m$,值组 $\langle f(\tilde{\alpha}^1), f(\tilde{\alpha}^2), \cdots, f(\tilde{\alpha}^m)\rangle$ 或者未完全有定义或者仍属于 G_m. P_k^* 中所有保 G_m 函数之集合记为 $T(G_m)$,称为**保 G_m 函数集**.

$T(G_m)$ 是封闭集,且包含自变数 x.

定理 20 保 E 函数集 T_E 是 P_k^* 的极大封闭集,$E \subset E_k$.

设 S_m 是集合 $E_m = \{1,2,\cdots,m\}$ 上的 m 次对称群,其单位元为 $e, \sigma \in S_m$. 令

$$G_m^{\sigma} = \{\langle a_{\sigma(1)}, a_{\sigma(2)}, \cdots, a_{\sigma(m)}\rangle \mid \langle a_1, \cdots, a_m\rangle \in G_m\},$$

容易证明,对任意 $\sigma \in S_m, T(G_m) = T(G_m^{\sigma})$.

设 H 是 S_m 的一个子群,称 m 项关系 G_m 是对 H 具有对称性或者说 H 是 G_m 的对称群,如果对任意 $\sigma \in H$,有 $G_m^{\sigma} = G_m$.

设 $H = \{\sigma \mid \sigma \in S_m, G_m^{\sigma} = G_m\}$,易证,$H$ 是 S_m 的一个子群,而且 H 是 G_m 的最大对称群. 规定空关系的最大对称群是 S_m.

设 $R_i \subseteq \{1,2,\cdots,m\}, i = 1,2,\cdots,n, |R_i| \geq 2$ 且当 $i \neq j$ 时,$R_i \cap R_j = \varnothing$. 令

$$G_m(R_1, R_2, \cdots, R_n) = \{\langle x_1, x_2, \cdots, x_m\rangle \mid x_i \in E_k, i = 1,2,\cdots,m \text{ 且当 } i,j \in R_h (1 \leq h \leq n) \text{ 时}, x_i = x_j\}$$

定义 27 称非完全关系 G_m 是**完满对称的**,如果

$$G_m = \bigcup_{\substack{i,j=1\\ i \neq j}}^{m} G_m(\{i,j\}) \cup G_m^*,$$

其中,G_m^* 是空集(当 $m = 2$ 时除外)或仅含 m 元不同的序列,且 G_m 对 S_m 是对称的. 所有保 G_m 函数之集合记为 $F_{s,m}$,称为**完满对称函数集**.

定理 21 $F_{s,m}$ 是极大封闭集.

令 $\overline{G}_m = \{\langle a_1, \cdots, a_m\rangle \mid a_i \neq a_j, i \neq j, a_i, a_j \in E_k, i, j = 1,2,\cdots,m\}$, $H = \{e, \sigma_1, \cdots, \sigma_{h-1}\}$.

定义 28 设 $G_m = \overline{G}_m \cup \overline{G}_m^{\sigma_1} \cup \cdots \cup \overline{G}_m^{\sigma_{h-1}}$,称 G_m 是**单纯可离的**,如果存在 E_k 的一个直接划分:

$$E_k = A_1 + \cdots + A_m, \quad A_i \cap A_j = \varnothing, i \neq j, i,j = 1,2,\cdots,m,$$

使得对任意 $\langle a_1, a_2, \cdots, a_m\rangle \in \overline{G}_m$ 皆有 $a_i \in A_i, i = 1,2,\cdots,m$. 若 G_m 单纯可离,则称 $T(G_m)$ 为**单纯可离函数集**,并记为 $S_{l,m}$.

定义 29 设 $G_m = G_m(R_1, R_2, \cdots, R_n) \cup G_m^*$, H 是 $G_m(R_1, R_2, \cdots, R_n)$ 的对称

群.$G_m^* = \overline{G}_m \cup \overline{G}_m^{\sigma_1} \cup \cdots \cup \overline{G}_m^{\sigma_{k-1}}$.不妨设

$$R_1 = \{1,2,\cdots,r_1\},\ R_2 = \{r_1+1,\cdots,r_2\},\ \cdots,\ R_n = \{r_{n-1}+1,\cdots,r_n\},$$

称 G_m 是**正则可离**的,如果存在一组直接划分:

$$D_i: E_k = A_1^{(i)} + \cdots + A_m^{(i)}, i = 1,2,\cdots,n,$$

使得对任意$\langle a_1,a_2,\cdots,a_m\rangle \in \overline{G}_m$必有一个具有下列性质的直接划分$D_h(1 \leqslant h \leqslant n)$:

$a_i \in A_i^{(h)}, i = 1,2,\cdots,m$,且当 $h \geqslant 2$ 时,有 $i_1,\cdots,i_{h-1},\cdots,j_1,\cdots,j_{h-1}$ 使

$a_1,\cdots,a_{r_1} \in A_{i_1}^{(1)} \cap A_{i_2}^{(2)} \cap \cdots \cap A_{i_{h-1}}^{(h-1)},\ \cdots,\ a_{r_{n-1}+1},\cdots,a_{r_n} \in A_{j_1}^{(1)} \cap A_{j_2}^{(2)} \cap \cdots \cap A_{j_{h-1}}^{(h-1)}$.

若 G_m 正则可离,则称 $T(G_m)$ 为**正则可离函数集**,记为 $S_{R,m}$.

定义 30 设 $G_4 = G_{4,3} \cup G_4^*$, $G_{4,3} = G_4(\{1,2\},\{3,4\}) \cup G_4(\{1,3\},\{2,4\}) \cup G_4(\{1,4\},\{2,3\})$, H 是$G_{4,3}$的最大对称群(即 S_4), $G_4^* = \overline{G}_4 \cup \overline{G}_4^{\sigma_1} \cup \cdots \cup \overline{G}_4^{\sigma_{k-1}}$.称 G_4 是**三度正则可离**的,如果存在一组直接划分:

$$D_i: E_k = A_1^{(i)} + \cdots + A_4^{(i)}, i = 1,2,\cdots,n,$$

使得对任意$\langle a_1,a_2,\cdots,a_4\rangle \in \overline{G}_4$必有一个具有下列性质的直接划分 $D_h(1 \leqslant h \leqslant n)$:

$a_i \in A_i^{(h)}, i = 1,2,\cdots,4$ 且对任意的 $D_j, 1 \leqslant j \leqslant h-1$,有 $A_{j_1}^{(j)}$、$A_{j_2}^{(j)}(1 \leqslant j_1,j_2 \leqslant 4)$ 使 a_1、a_2(或 a_1、a_3 或 a_1、a_4) $\in A_{j_1}^{(j)}$, a_3、a_4(或 a_2、a_4 或 a_2、a_3) $\in A_{j_2}^{(j)}$.

如果 G_4 是三度正则可离的,或 G_4^* 是空集,则称 $T(G_4)$ 为**拟线性函数集**,并记为 L_p.

显然,当 E_k 是 2^n 阶初等加法群时, P_k 中的任意线性函数集必包含在某一个拟线性函数集中.

定义 31 设 $G_4 = G_{4,2} \cup G_4^*$, $G_{4,2} = (\{1,2\},\{3,4\}) \cup (\{1,3\},\{2,4\})$, H 是$G_{4,2}$的最大对称群,即 S_4, $G_4^* = \overline{G}_4 \cup \overline{G}_4^{\sigma_1} \cup \cdots \cup \overline{G}_4^{\sigma_{k-1}}$.称 G_4 是**二度正则可离**的,如果存在一组直接划分:

$$D_i: E_k = A_1^{(i)} + \cdots + A_4^{(i)}, i = 1,2,\cdots,n,$$

使得对任意$\langle a_1,a_2,\cdots,a_4\rangle \in \overline{G}_4$必有一个具有下列性质的直接划分 $D_h(1 \leqslant h \leqslant n)$:

$a_i \in A_i^{(h)}, i = 1,2,\cdots,4$ 且对任意的 $D_j, 1 \leqslant j \leqslant h-1$,有 $A_{j_1}^{(j)}$、$A_{j_2}^{(j)}(1 \leqslant j_1$、$j_2 \leqslant 4)$ 使

a_1、a_2(或 a_1、a_3) $\in A_{j_1}^{(j)}$, a_3、a_4(或 a_2、a_4) $\in A_{j_2}^{(j)}$.

如果 G_4 是二度正则可离的,或 G_4^* 是空集,则称 $T(G_4)$ 为 **L 型函数集**,并记为 $L_{G_{4,2}}$.

定理 22 单纯可离函数集 $S_{I,m}$、正则可离函数集 $S_{R,m}$、拟线性函数集 L_p、L 型函数集 $L_{G_{4,2}}$ 都是极大封闭集.

3.2.3　完备性定理

定理 23　P_k^*（$k \geqslant 3$）中的全部极大封闭集是：$P_k \cup \{*\}$、完满对称函数集 $F_{s,m}$、拟线性函数集 L_p、L 型函数集 $L_{G_{4,2}}$、单纯可离函数集 $S_{I,m}$、正则可离函数集 $S_{R,m}$ 及保 E 函数集 T_E.

定理 24(完备性定理)　设 $A \subseteq P_k^*$，A 非空，$k \geqslant 3$，则 A 是完备集，当且仅当 A 不包含在极大封闭集：$P_k \cup \{*\}$、$F_{S,m}$、L_p、$L_{G_{4,2}}$、$S_{I,m}$、$S_{R,m}$ 及 T_E 的任何一个之中.

根据完备性定理，刘任任证明了下述定理：

定理 25　P_3^* 中的函数 f 为谢弗函数的充要条件是 f 不属于下列 26 个极大封闭集的任何一个之中：

(1) 保 E 函数集 T_E 共 6 个，$T_{\{0\}}$，$T_{\{1\}}$，$T_{\{2\}}$，$T_{\{0,1\}}$，$T_{\{0,2\}}$，$T_{\{1,2\}}$.

(2) 完满对称函数集共 4 个，$F_{S,m} = T(G_m)$，$2 \leqslant m \leqslant 3$，

1) $m = 2$ 时，$G_2 = \{\langle 0,0\rangle,\langle 1,1\rangle,\langle 2,2\rangle\} \cup G_2^*$，$G_2^*$ 为下列之一：

$\{\langle 0,1\rangle,\langle 1,0\rangle\}$，$\{\langle 0,2\rangle,\langle 2,0\rangle\}$，$\{\langle 1,2\rangle,\langle 2,1\rangle\}$

2) $m = 3$ 时，$G_3 = G_3(\{1,2\}) \cup G_3(\{1,3\}) \cup G_3(\{2,3\})$.

(3) 单纯可离函数集共 8 个，$S_{I,m} = T(G_m)$，

1) $m = 2$ 时有 6 个，$S_{I,2} = T(G_2)$，G_2 为下列之一：

$\{\langle 0,1\rangle,\langle 1,0\rangle\}$，$\{\langle 0,2\rangle,\langle 2,0\rangle\}$，$\{\langle 1,2\rangle,\langle 2,1\rangle\}$，$\{\langle 0,1\rangle,\langle 1,0\rangle,\langle 0,2\rangle,\langle 2,0\rangle\}$，$\{\langle 0,2\rangle,\langle 2,0\rangle,\langle 1,2\rangle,\langle 2,1\rangle\}$，$\{\langle 0,1\rangle,\langle 1,0\rangle,\langle 1,2\rangle,\langle 2,1\rangle\}$.

2) $m = 3$ 时有 2 个 $S_{I,3} = T(G_3)$，G_3 是 $\{a_0,a_3,a_4\}$ 或者 $\{a_0,a_1,a_2,a_3,a_4,a_5\}$，其中，$a_i$（$i = 0,\cdots,5$）是轮换，且 $a_0 = (012)$，$a_1 = (021)$，$a_2 = (102)$，$a_3 = (120)$，$a_4 = (201)$，$a_5 = (210)$

(4) 正则可离函数集共 5 个，$S_{R,m} = T(G_m)$，

1) $m = 2$ 时有 3 个，$S_{R,2} = T(G_2)$，其中 $G_2 = \{\langle 0,0\rangle,\langle 1,1\rangle,\langle 2,2\rangle\} \cup G_2^*$，$G_2^*$ 是下列之一：$\{\langle 0,1\rangle\}$，$\{\langle 0,2\rangle\}$，$\{\langle 1,2\rangle\}$.

2) $m = 3$ 时有 2 个，$S_{R,3} = T(G_3)$，其中 $G_3 = \{\langle 0,0,0\rangle,\langle 1,1,1\rangle,\langle 2,2,2\rangle\} \cup G_3^*$，$G_3^*$ 为 $\{a_0,a_3,a_4\}$ 或者 $\{a_0,a_1,a_2,a_3,a_4,a_5\}$.

(5) 拟线性函数集 $L_p = T(G_{4,3})$.

(6) L 型函数集 $L_{G_{4,2}} = T(G_{4,2})$.

(7) $P_3 \cup \{*\}$.

定理 26　以下四类极大封闭集必为 P_k^*（$k \geqslant 3$）中所有极大封闭集之最小覆盖的组成部分：

(1) 保 E 函数集 T_E.

(2) 拟线性函数集 L_p.

(3) L 型函数集 $L_{G_{4,2}}$.

(4) $P_k \cup \{*\}$.

3.3　一元多值逻辑函数

一元多值逻辑函数系完备性的判定依赖于定出 S_k 中的所有极大子群.但在有限群论中,定出 S_k 的所有极大子群至今还是一个尚未解决的问题.

设 $P_{k,1}^*$ 是 P_k^* 中所有一元函数作成之集合;Q_n 是 $P_{k,1}^*$ 中所有至多取 n 个值的完全函数作成之集合,$n \leqslant k-1$;Q_n^* 为 $P_{k,1}^*$ 中所有至多取 n 个值的非完全函数作成之集合,$n \leqslant k-1$;S_k 是 E_k 上的 k 次对称群,$E_k = \{0,1,\cdots,k-1\}$.

定理 27　$P_{k,1}^*$ 中的极大封闭集必是下列一元极大封闭集:

$$S_k \cup Q_{k-1} \cup Q_{k-2}^*, S_k \cup Q_{k-2} \cup Q_{k-1}^*, M_i \cup Q_{k-1} \cup Q_{k-1}^*, \quad i = 1,\cdots,m,$$

其中 $M_1, M_2, \cdots, M_m$ 是 S_k 中的所有极大子群.

定理 28　$P_{k,1}$ 中的极大封闭集必是下列一元极大封闭集:

$$S_k \cup Q_{k-2}, M_i \cup Q_{k-1}, \quad i = 1,\cdots,m,$$

其中 $M_1, M_2, \cdots, M_m$ 是 S_k 中的所有极大子群.

由定理1和定理2,只要定出 S_k 中的所有极大子群便可得到 $P_{k,1}^*$ 和 $P_{k,1}$ 中的全部极大封闭集.

定义 32　设置换群 $G \subseteq S_k, E \subseteq E_k$.称 E 是 G 的一个不动块,如果对任意 $g \in G$ 有 $E^g = E$,其中 $E^g = \{a^g \mid a \in E\}$.

显然,E_k、$\varnothing$ 是不动块,称为**平凡不动块**.

定义 33　称 $G(\subseteq S_k)$ 说是**可迁的**,如果 G 仅有平凡的不动块;否则称为**不可迁的**.

显然,G 是可迁的充要条件是,对任意 a、$b \in E_k$ 有 $g \in G$ 使 $a^g = b$($a^g = g(a)$).

定理 29　置换群分为下列互不相同的5类:

1° 非可迁群;

2° 非本原群;

3° 仿射群 $AGL(n,p)$,$k = p^n$,p 是质数;

4° 保正则2项关系的置换群;

5° 基本置换群.

只要定出以上5类的极大子群,便可得到 S_k 的全部极大子群.

定理 30　设 G 是一个不可迁群,$G \subseteq S_k$,则 G 是 S_k 的极大子群的充要条件是有直接划分

$$D: E_k = A + B, \quad |A| \neq |B|,$$

使得 $G = S_A \cdot S_B$,其中 S_A、S_B 分别是 A、B 上的对称群.

定义 34　设置换群 $G \subseteq S_k, E \subseteq E_k$. E 说是 G 的一个块,如果对任意 $g \in G$ 有 $E^g = E$ 或 $E^g \cap E = \varnothing$.

显然,E_k、$\varnothing$、$\{a\}(a \in E_k)$ 是块,称为**平凡块**.

定义 35　设 $G(\subseteq S_k)$ 是一个可迁群,若 G 仅有平凡块,则称 G 是**本原的**;否则

称为非本原的.

定理 31 设 G 是一个非本原群，$G \subsetneq S_k$. 则 G 是 S_k 的极大子群的充要条件是有直接划分

$$D: E_k = A_1 + \cdots + A_m, \quad |A_1| = \cdots = |A_m|, 1 < m < k,$$

使得 G 是保划分 D 的置换之集合.

定理 32 设 E_k 是一个加法群，$k \geqslant 5$. 则 $\mathrm{AGL}(n,p)(p=2)$ 是 k 次交代群 A_k 的极大子群，当且仅当 E_k 是一个 2^n 阶的初等加法群.

设 E_k 是一个加法群，$k \geqslant 5$，则 $\mathrm{AGL}(n,p)(p=2)$ 是 k 次对称群 S_k 的极大子群，当且仅当 E_k 是一个 p^n 阶的初等加法群，$p \neq 2$(p 是质数).

定理 33 当 $h \geqslant 3$ 时，保正则二项关系置换群 H_{R,h^m} 是本原群且 $g \in H_{R,h^m}$ 的充要条件是有 $g_i \in S_h, i = 1,2,\cdots,m$ 与 $\sigma \in S_m$，使得 $g(x_1,x_2,\cdots,x_m) = (g_1(x_{\sigma(1)}),\cdots,g_m(x_{\sigma(m)}))$. 其中 $x_i \in E_h, i = 1,2,\cdots,m, k = h^m, m \geqslant 2$.

定理 34 (1) 设 $h \geqslant 6$，则当 h 为奇数或 $h = 2n$，n 为奇数时，H_{R,h^2} 是 S_{h^2} 的极大子群；当 $h = 2n$，n 为偶数时，H_{R,h^2} 是 A_{h^2} 的极大子群，且 H_{R,h^2} 与 S_{h^2} 之间除 A_{h^2} 外无其它子群.

(2) 设 $k = h^m, h \geqslant 5, m \geqslant 3$. 则当 h 为奇数时，H_{R,h^m} 是 S_k 的极大子群；当 h 为偶数时，H_{R,h^m} 是 A_k 的极大子群，且 H_{R,h^m} 与 S_k 之间除 A_k 外无其它子群.

只要定出全部的极大基本群，便得到 S_k 的全部极大子群.

定理 35 设 p 是质数，$p \geqslant 5$，则 p 次对称群 S_p 的全部极大子群是下列 3 类：

1° p 次交代群 A_p；

2° 线性置换群 $L(E_p)$；

3° 不可迁群 $S_A \cdot S_B$，其中 S_A、S_B 分别是子集合 A、B 上的对称群；$E_p = A + B$，$A \cap B = \varnothing$，$|A| \neq |B|$.

根据定理 9 可计算出 S_p 中的全部极大子群的个数为 $2^{p-1} + (p-2)!$.

目前已分别定出了前 4 类置换群在 S_k 中的所有极大子群，对于第 5 类置换群在 S_k 中的极大子群，目前还只有一些部分结果，离问题的解决还有较大的距离. 但由于它们与多值逻辑函数密切相关，因此可以预见，在其极大性的研究中，多值逻辑函数的结构理论必将成为有力的工具之一.

参 考 文 献

1 罗铸楷，胡谋，陈廷槐著. 多值逻辑的理论及应用. 北京：科学出版社，1992.

2 David C. Rine. Computer science and multiple-valued logic theory and applications. 2ed. North-Holland: Elsevier Science Publishers BV, 1984.

3 王雨田主编. 现代逻辑科学导引. 北京：中国人民大学出版社，1987.

索 引

使用说明：1. 本索引收录了本卷正文中用黑体排印的大部分术语。

2. 术语依第一字的读音按汉语拼音字母表顺序排列。如果拼音相同，根据音调，按阴平、阳平、上声、去声、轻声的次序排列。如果音和音调也相同，按总笔画数排列。

3. 以符号、数字或字母起首的术语，按符号、数字、拉丁字母、希腊字母的顺序，分别集中排在以汉字起首的术语前面，其中字母依首写字母按字母的顺序排列。

4. 以数学家译名为首的术语（例如，傅里叶变换），依译名按汉字的排法排列。

5. 术语后面的数字，表示该术语出现在本书中的页码。

以符号、数字起首的术语

以拉丁字母起首的术语

以汉字起首的术语

A

B

C

D

E

F

G

H

J

K

L

R

S

T

图书在版编目(CIP)数据

现代数学手册·计算机数学卷/《现代数学手册》编纂委员会
武汉:华中科技大学出版社,2001年2月
ISBN 7-5609-2174-4

Ⅰ.现…
Ⅱ.现…
Ⅲ.①数学-手册 ②电子计算机-数学-手册
Ⅳ.O 1-62

现代数学手册·计算机数学卷 《现代数学手册》编纂委员会

责任编辑:龙纯曼 余健棠 封面设计:刘 卉
责任校对:张 欣 责任监印:张正林

出版发行:华中科技大学出版社 武昌喻家山 邮编:430074 电话:(027)87545012
经销:新华书店湖北发行所

录排:湖北省新华印刷厂
印刷:湖北省新华印刷厂

开本:880×1230 1/32 印张:34.5 插页:6 字数:1 340 000
版次:2001年2月第1版 印次:2001年2月第1次印刷 印数:1—8 000
ISBN 7-5609-2174-4/O·207 定价:100.00元